Electronic Transactions on Numerical Analysis.
Volume 44, pp. 189–229, 2015.
Copyright © 2015, Kent State University.
ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

FAST SOLUTION OF BOUNDARY INTEGRAL EQUATIONS WITH THE GENERALIZED NEUMANN KERNEL*

MOHAMED M. S. NASSER†

Abstract. A fast method for solving boundary integral equations with the generalized Neumann kernel and the adjoint generalized Neumann kernel is presented. The complexity of the developed method is $O((m+1)n\ln n)$ for the integral equation with the generalized Neumann kernel and $O((m+1)n)$ for the integral equation with the adjoint generalized Neumann kernel, where $m+1$ is the multiplicity of the multiply connected domain and n is the number of nodes in the discretization of each boundary component. Numerical results illustrate that the method gives accurate results even for domains of very high connectivity, domains with piecewise smooth boundaries, domains with close-to-touching boundaries, and domains of real world problems.

Key words. generalized Neumann kernel, boundary integral equations, Nyström method, Fast Multipole Method, GMRES, numerical conformal mapping

AMS subject classifications. 45B05, 65R20, 30C30

1. Introduction.

Let G be an $(m+1)$-multiply connected domain in the extended complex plane $\mathbb{C}\cup\{\infty\}$. Let G have the boundary

$$\Gamma := \partial G = \bigcup_{j=0}^{m} \Gamma_j\,,$$

where $\Gamma_0, \Gamma_1, \ldots, \Gamma_m$ are closed Jordan curves. The orientation of Γ is such that G is always on the left of Γ. The domain G can be bounded or unbounded. For the case when G is bounded, we assume that α is a given point in G and Γ_0 encloses all the other curves $\Gamma_1, \ldots, \Gamma_m$. If G is unbounded, then $\infty \in G$. For $m = 0$, the domain G is simply connected; see Figure 1.1.

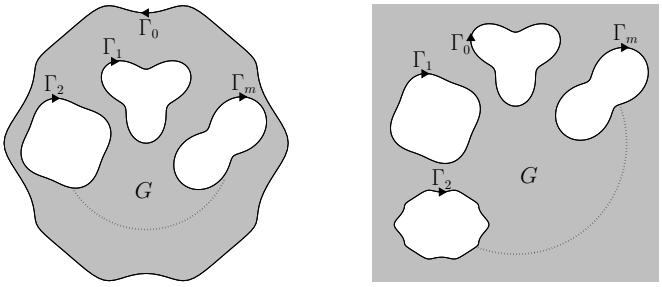

FIG. 1.1. *The bounded (left) and unbounded (right) multiply connected domain G of connectivity $m+1$.*

Consider the following classical boundary value problem for analytic functions:

The Riemann-Hilbert problem. Let A be a Hölder continuous complex-valued function on Γ with $A \neq 0$ and γ be a Hölder continuous real-valued function on Γ. Find a function f analytic in G, with $f(\infty) = 0$ for unbounded G, and continuous on the closure $\overline{G}$ such that the boundary values of f satisfy on Γ:

$$\mathrm{Re}[Af] = \gamma \qquad \text{on } \Gamma. \tag{1.1}$$

*Received March 28, 2014. Accepted February 12, 2015. Published online on April 8, 2015. Recommended by M. Donatelli.

†Department of Mathematics, Faculty of Science, King Khalid University, P. O. Box 9004, Abha 61413, Saudi Arabia (mmnasser@kku.edu.sa, mms_nasser@hotmail.com).

ETNA
Kent State University
http://etna.math.kent.edu

When $A = 1$, problem (1.1) is also known as the modified Dirichlet problem [11, 27, 30] and the Schwarz problem [9, 11].

The solvability of the Riemann-Hilbert problem (1.1) depends upon the index (winding number in other terminology) of the function A [11, 50]. The index κ_j of the function A on the curve Γ_j is defined as the change of the argument of A along the curve Γ_j divided by 2π, i.e.,

$$\kappa_j := \frac{1}{2\pi} \Delta \arg A|_{\Gamma_j}, \qquad j = 0, 1, \ldots, m.$$

The index κ of the function A on the whole boundary Γ is the sum of the indexes κ_j, i.e., $\kappa = \sum_{j=0}^{m} \kappa_j$.

In this paper, we assume that the function $A(\eta)$ is defined for $\eta \in \Gamma$ by

$$A(\eta) = \begin{cases} e^{\mathrm{i}\left(\frac{\pi}{2} - \theta(\eta)\right)} (\eta - \alpha), & \text{if } G \text{ is bounded,} \\ e^{\mathrm{i}\left(\frac{\pi}{2} - \theta(\eta)\right)}, & \text{if } G \text{ is unbounded,} \end{cases} \tag{1.2}$$

where θ is a piecewise constant real-valued function defined on Γ, i.e.,

$$\theta(\eta) = \theta_j, \qquad \text{for } \eta \in \Gamma_j, \tag{1.3}$$

and θ_j are given real constants for $j = 0, 1, \ldots, m$. For simplicity, the piecewise constant function $\theta(\eta)$ defined on Γ by (1.3) will be denoted by

$$\theta(\eta) = (\theta_0, \theta_1, \theta_2, \ldots, \theta_m). \tag{1.4}$$

This notation will be used for any piecewise constant function defined on Γ.

The index of the function A in (1.2) is given for bounded G by

$$\kappa_0 = 1, \quad \kappa_1 = \cdots = \kappa_m = 0, \quad \kappa = 1,$$

and for unbounded G by

$$\kappa_0 = \kappa_1 = \cdots = \kappa_m = 0, \quad \kappa = 0.$$

Thus, the Riemann-Hilbert problem (1.1) with A in (1.2) is not necessary solvable [11, 33, 50, 53, 54]. However, a unique piecewise constant real-valued function $h(\eta) = (h_0, h_1, \ldots, h_m)$ exists such that the following Riemann-Hilbert problem

$$\mathrm{Re}[Af] = \gamma + h \qquad \text{on } \Gamma \tag{1.5}$$

is uniquely solvable [31, 32, 35, 36]. The unknown function h is part of the solution of problem (1.5) and should be calculated alongside the function f. In the next section, we shall review a method for solving the Riemann-Hilbert problem (1.5). The method is based on a boundary integral equation with the generalized Neumann kernel.

The function $\theta(\eta)$ in (1.2) gives us the flexibility to reformulate several problems in applied mathematics as problem (1.5) with the function A in (1.2). For example, the three classical boundary value problems: the Dirichlet problem, the Neumann problem, and the mixed Dirichlet-Neumann problem can be reduced to (1.5) [2, 3, 9, 11, 16, 17, 27, 28, 30, 41, 42]. More importantly, the problem of computing the conformal mapping onto Koebe's thirty-nine canonical slit domains (see Figures 1–39 in [21]) has been rephrased into this form [31, 32, 35, 36]. The same holds for the computation of the conformal mapping onto the canonical domain obtained by removing several slits from a strip; see [54, p. 128]. Furthermore, the problem of computing the potential flow past multiple aerofoils can also be reduced to (1.5) [34]; see also [7, 8]. Several other tasks in mathematical physics can be formulated as a Riemann-Hilbert problem (1.5); see, e.g., [45]. Table 1.1 demonstrates how to choose the function $\theta(\eta)$ in each of these cases.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 1.1
The choices of the function $\theta(\eta) = (\theta_0, \theta_1, \theta_2, \ldots, \theta_m)$ for different problems.

$\theta(\eta)$	Applications
$\theta_j = \pi/2$, for $j = 0, 1, \ldots, m$	• the Dirichlet problem (see [42]) • the Neumann problem (see [42]) • the conformal mapping onto the canonical domains in Figures 1, 2, and 3 in Koebe [21] (see [31, 32, 43])
$\theta_j \in \{0, \pi/2\}$, for $j = 0, 1, \ldots, m$	• the mixed Dirichlet-Neumann problem [2, 3, 41] • the conformal mapping onto the canonical domains in Figures 8 and 9 in Koebe [21] (see [35])
$\theta_j = 0$, for $j = 0, 1, \ldots, m$	• the conformal mapping onto the canonical domains in Figures 4, 5, 14, 15, 16, 20, 21, 26, 27, 28, 29, 34, 35, and 36 in Koebe [21] (see [31, 32, 36, 43]) • the conformal mapping onto a strip with a parallel slits (see [39]) • the external potential flow problem (see [34])
$\theta_0 = \pi/2$, $\theta_j \in \{0, \pi/2\}$, for $j = 1, \ldots, m$	• the conformal mapping onto the canonical domains in Figure 7 in Koebe [21] (see [35])
$\theta_0 = 0$, $\theta_j \in \{0, \pi/2\}$, for $j = 1, \ldots, m$	• the conformal mapping onto the canonical domains in Figures 17, 18, 19, 22, 24, 30, 31, 32, and 33 in Koebe [21] (see [36])
$\theta_0 = \pi/2$, $\theta_1 = \pi/2$, $\theta_j \in \{0, \pi/2\}$, for $j = 2, \ldots, m$	• the conformal mapping onto the canonical domains in Figure 6 in Koebe [21] (see [35])
$\theta_0 = 0, \theta_1 = 0$, $\theta_j \in \{0, \pi/2\}$, for $j = 2, \ldots, m$	• the conformal mapping onto the canonical domains in Figures 37, 38, and 39 in Koebe [21] (see [36])
$\theta_0 = \pi/2, \theta_1 = 0$, $\theta_j \in \{0, \pi/2\}$, for $j = 2, \ldots, m$	• the conformal mapping onto the canonical domains in Figures 23 and 25 in Koebe [21] (see [36])
$\theta_j \in [0, \pi/2]$, for $j = 0, 1, \ldots, m$	• the conformal mapping onto the canonical domains in Figures 10, 11, 12, and 13 in Koebe [21] (see [36])

2. The generalized Neumann kernel. For $j = 0, 1, \ldots, m$, the curve Γ_j is parametrized by a 2π-periodic twice continuously differentiable complex function $\eta_j(t)$ with non-vanishing first derivative $\dot{\eta}_j(t) \neq 0$, for $t \in J_j := [0, 2\pi]$. (A dot always denotes the derivative with respect to the parameter t.) The total parameter domain J is the disjoint union of the $m + 1$ intervals $J_0, J_1, \ldots, J_m$,

$$J = \bigsqcup_{j=0}^{m} J_j = \bigcup_{j=0}^{m} \{(t, j) \ : \ t \in J_j\},$$

ETNA
Kent State University
http://etna.math.kent.edu

i.e., the elements of J are order pairs (t, j), where j is an auxiliary index indicating which of the intervals contains the point t [24, p. 394]. We define a parametrization of the whole boundary Γ as the complex function η defined on J by

$$\eta(t, j) = \eta_j(t), \qquad t \in J_j, \quad j = 0, 1, \ldots, m. \tag{2.1}$$

In this paper, we assume that for a given t the auxiliary index j is known, so we replace the pair (t, j) on the left-hand side of (2.1) by t. Thus, the function η in (2.1) is written as

$$\eta(t) = \begin{cases} \eta_0(t), & t \in J_0, \\ \eta_1(t), & t \in J_1, \\ \quad \vdots & \\ \eta_m(t), & t \in J_m. \end{cases}$$

In view of the smoothness of the parametrization $\eta(t)$ of the boundary Γ, any real-valued or complex-valued function $\phi(\eta)$, Hölder continuous on the boundary Γ, can be interpreted via $\hat{\phi}(t) := \phi(\eta(t))$ as a 2π-periodic Hölder continuous function of the parameter t on J and vice versa. Henceforth, in this paper, we shall not distinguish between $\phi(t)$ and $\phi(\eta(t))$.

The generalized Neumann kernel formed with A and η is defined on $J \times J$ by [29, 52, 53]

$$N(s, t) := \frac{1}{\pi} \operatorname{Im}\left(\frac{A(s)}{A(t)} \frac{\dot{\eta}(t)}{\eta(t) - \eta(s)}\right).$$

Another kernel which is always defined together with $N(s, t)$ is the kernel $M(s, t)$ defined on $J \times J$ by

$$M(s, t) := \frac{1}{\pi} \operatorname{Re}\left(\frac{A(s)}{A(t)} \frac{\dot{\eta}(t)}{\eta(t) - \eta(s)}\right).$$

LEMMA 2.1 ([53]).
(a) The kernel N is continuous with

$$N(t, t) = \frac{1}{\pi}\left(\frac{1}{2} \operatorname{Im} \frac{\ddot{\eta}(t)}{\dot{\eta}(t)} - \operatorname{Im} \frac{\dot{A}(t)}{A(t)}\right).$$

(b) When $s, t \in J_j$ are in the same parameter interval J_j, then

$$M(s, t) = -\frac{1}{2\pi} \cot \frac{s - t}{2} + M_1(s, t) \tag{2.2}$$

with a continuous kernel M_1, which takes the values on the diagonal

$$M_1(t, t) = \frac{1}{\pi}\left(\frac{1}{2} \operatorname{Re} \frac{\ddot{\eta}(t)}{\dot{\eta}(t)} - \operatorname{Re} \frac{\dot{A}(t)}{A(t)}\right).$$

Let H be the space of all real-valued Hölder continuous functions on the boundary Γ. On H we define the following operators:

$$\mathbf{N}\mu(s) := \int_J N(s, t)\mu(t)dt, \qquad s \in J,$$
$$\mathbf{M}\mu(s) := \int_J M(s, t)\mu(t)dt, \qquad s \in J.$$

ETNA
Kent State University
http://etna.math.kent.edu

In view of the previous lemma, the integral operator $\mathbf{N}$ is a compact operator, and the operator $\mathbf{M}$ is a singular operator. Both operators $\mathbf{N}$ and $\mathbf{M}$ are bounded on the space H and map H into itself. For more details, see [33, 52, 53].

The adjoint operator $\mathbf{N}^*$ is defined by

$$\mathbf{N}^*\mu(s) := \int_J N^*(s,t)\mu(t)dt, \qquad s \in J,$$

where $N^*(s,t) = N(t,s)$. We also define an integral operator $\mathbf{J}$ by

$$\mathbf{J}\mu(s) := \int_J \delta(s,t)\mu(t)dt, \qquad s \in J,$$

where the kernel $\delta(s,t)$ is given for $s \in J_k$ and $t \in J_j$ by

$$\delta(s,t) = \begin{cases} \frac{1}{2\pi}, & k = j, \\ 0, & k \neq j, \end{cases}$$

for $k, j = 0, 1, \ldots, m$. Hence,

$$\mathbf{J}\mu(s) = \left(\frac{1}{2\pi}\int_{J_0} \mu(t)dt, \frac{1}{2\pi}\int_{J_1} \mu(t)dt, \ldots, \frac{1}{2\pi}\int_{J_m} \mu(t)dt \right),$$

i.e., the function $\mathbf{J}\mu(s)$ is piecewise constant.

To solve the Riemann-Hilbert problem (1.5), it suffices to find the boundary values of its unique solution f on the boundary Γ. Then the values of $f(z)$, for $z \in G$, can be computed by the Cauchy integral formula.

Let $\mu = \mathrm{Im}[Af]$. The boundary values of the function f are given by

$$Af = \gamma + h + \mathrm{i}\mu,$$

where only the function γ is known and the functions h and μ are unknowns. The unknown functions h and μ are uniquely determined by the known γ. Both functions μ and h can be determined by solving an integral equation with the generalized Neumann kernel as described in the following theorem from [31, 32, 35].

THEOREM 2.2. *For any $\gamma \in H$, there exists a unique function $\mu \in H$ and a unique piecewise constant real-valued function $h = (h_0, h_1, \ldots, h_m)$ such that*

$$A(\eta)f(\eta) = \gamma(\eta) + h(\eta) + \mathrm{i}\mu(\eta), \qquad \eta \in \Gamma,$$

are the boundary values of the unique solution of the Riemann-Hilbert problem (1.5). The function μ is the unique solution of the integral equation

$$(\mathbf{I} - \mathbf{N})\mu = -\mathbf{M}\gamma\,, \tag{2.3}$$

and the function h is given by

$$h = [\mathbf{M}\mu - (\mathbf{I} - \mathbf{N})\gamma]/2. \tag{2.4}$$

It is clear that the unique solution μ of (2.3) yields the unknown piecewise constant function h in (1.5) and the boundary values of the unique solution $f(z)$ of the Riemann-Hilbert problem (1.5). The values of the function $f(z)$ for interior points $z \in G$ can be computed

ETNA
Kent State University
http://etna.math.kent.edu

by the Cauchy integral formula. Hence, the integral equation (2.3) can be used to solve the problems listed in Table 1.1.

The main objective of this paper is to develop a fast numerical method for solving the boundary integral equation with the generalized Neumann kernel (2.3) and computing the piecewise constant function h given by (2.4). We shall also develop a fast numerical method for solving the boundary integral equation with the adjoint generalized Neumann kernel

$$(\mathbf{I} + \mathbf{N}^* + \mathbf{J})\mu = \gamma\,, \tag{2.5}$$

where $\gamma \in H$. Similar to equation (2.3), the boundary integral equation (2.5) can be used to compute the conformal mapping onto Koebe's thirty-nine canonical slit domains [43, 55, 56] and to solve the potential flow problem [34, 45].

As for the Riemann-Hilbert problem, the solvability of integral equations with the operators $\mathbf{N}$ and $\mathbf{N}^*$ depends on the function A. For A given by (1.2), the integral equations (2.3) and (2.5) are uniquely solvable. For more details, see [29, 33, 52, 53].

THEOREM 2.3 ([42]). *Suppose that θ is a constant function and the function A is given by* (1.2).

(a) If λ is an eigenvalue of $\mathbf{I} - \mathbf{N}$, then $\lambda \in (0, 2]$.

(b) If λ is an eigenvalue of $\mathbf{I} + \mathbf{N}^* + \mathbf{J}$, then $\lambda \in (0, 2)$.

Since the operator $\mathbf{N}$ and its adjoint $\mathbf{N}^*$ are compact, the eigenvalues of $\mathbf{I} - \mathbf{N}$ and $\mathbf{I} + \mathbf{N}^*$ are clustered at 1 [23, p. 292]. It follows from [42] that if λ is an eigenvalue of $\mathbf{I} + \mathbf{N}^* + \mathbf{J}$, then λ is also an eigenvalue of $\mathbf{I} + \mathbf{N}^*$. Hence, the eigenvalues of the operator $\mathbf{I} + \mathbf{N}^* + \mathbf{J}$ are also clustered at 1. As stated in Theorem 2.3, the eigenvalues of $\mathbf{I} - \mathbf{N}$ and $\mathbf{I} + \mathbf{N}^* + \mathbf{J}$ are real for constant functions θ. For non-constant θ, numerical evidence shows that the operators $\mathbf{I} - \mathbf{N}$ and $\mathbf{I} + \mathbf{N}^* + \mathbf{J}$ have complex eigenvalues and these eigenvalues are in the closed disk centred at 1 with radius 1; see [39, 40] and the numerical results below.

For bounded or unbounded multiply connected domains of connectivity $m+1$, discretizing the boundary integral equations (2.3) and (2.5) by the Nyström method with the trapezoidal rule yields dense and nonsymmetric $(m + 1)n \times (m + 1)n$ linear systems, where n is the number of nodes in the discretization of each boundary component. If the boundaries are of class C^{q+2} and the function γ is of class C^q, then the rate of convergence of the Nyström method with the trapezoidal rule is $O(1/n^q)$. For analytic boundaries and analytic function γ, the Nyström method with the trapezoidal rule converges exponentially [22]. For domains with corners, accurate results are obtained if we use the trapezoidal rule with a graded mesh [22, 23].

The condition number of the obtained linear systems is bounded independently of n [23, 39, 42]. Thus, if the generalized minimal residual (GMRES) method [49] is used to solve these systems, the number of iterations required for a certain accuracy is expected to be bounded independently of n. Hence, the computational complexity of the GMRES method is determined by the amount of work required for computing matrix-vector products. An elegant method for a fast and efficient computation of matrix-vector products is the Fast Multipole Method (FMM) developed by Rokhlin [48] and Greengard and Rokhlin [15]. For more details, see [1] and [23, §14.4]. See also [12, 14, 18, 19, 25, 46, 48] for examples of integral equations which have been solved by the FMM.

Recently, a fast method for solving the $(m + 1)n \times (m + 1)n$ linear systems resulting from the discretization of (2.3) has been presented in [39], where the linear system is solved by the GMRES method. Each iteration of the GMRES method requires a matrix-vector product, which can be computed using the FMM in $O((m + 1)n)$ operations. However, the discretization of the singular operator $\mathbf{M}$ in (2.3) and (2.4) requires $O((m + 1)n \ln n)$ operations. Discretizing the operator $\mathbf{M}$ in [39] requires computing the derivatives $\dot{\gamma}$ and $\dot{\mu}$, where the derivative of the known function γ can be computed analytically and the derivative of the unknown function μ should be computed numerically.

ETNA
Kent State University
http://etna.math.kent.edu

This paper presents a new method for the fast computation of $\mathbf{M}\gamma$ in (2.3) and $\mathbf{M}\mu$ in (2.4). We assume that the functions γ and μ are only Hölder continuous without any further differentiability requirement. Thus, the stability issue of the numerical differentiation of the function μ is avoided. We shall rewrite the discretizing matrix of the operator $\mathbf{M}$ as a sum of two matrices. The multiplication of the first matrix by a vector can be computed by the FMM in $O((m+1)n)$ operations. The second matrix is a block of $m+1$ circulant matrices. Hence, the multiplication of the second matrix by a vector can be done by the FFT in $O((m+1)n\ln n)$ operations. Thus, as in [39], the discretized linear system is solved by a combination of the GMRES method and the FMM in $O((m+1)n)$ operations. Consequently, the unique solution μ of the integral equation (2.3) and h in (2.4) are computed in $O((m+1)n\ln n)$ operations. In addition to the fast method for solving the integral equation (2.3), this paper also gives a new method for the fast solution of the integral equation with the adjoint generalized Neumann kernel (2.5) in $O((m+1)n)$ operations. Based on the discussed methods, two MATLAB functions will be presented in this paper:

1. `fbie`: a function for fast solution of the integral equation with the generalized Neumann kernel (2.3) and fast computation of the piecewise constant function h in (2.4).
2. `fbiead`: a function for fast solution of the integral equation with the adjoint generalized Neumann kernel (2.5).

The solutions of the integral equations (2.3) and (2.5) yield the boundary values of the conformal mapping and the solutions of the boundary value problems. Computing the interior values requires computing the Cauchy integral formula. For the convenience of the reader, we present a MATLAB function `fcau` for the fast computation of the Cauchy integral formula. The MATLAB functions, `fbie`, `fbiead`, and `fcau`, will be useful for calculating the conformal mapping and solving potential flow problems for domains of high connectivity.

In this paper, the performance of the presented method will be tested on five numerical examples, which include various types of multiply connected domains. Recently, the method proposed in this paper has been used in [38] for fast computation of conformal mappings onto circular domains, in [45] for fast computation of potential flows in multiply connected coastal domains, and in [37] for fast computation of conformal mappings between simply connected domains. The numerical results in this paper together with those in [37, 38, 45] verify that the presented method can be used for a wide range of problems involving multiply connected domains.

3. The MATLAB function `zfmm2dpart`. Let n be a given even positive integer. For $k = 0, 1, \ldots, m$, we define in each interval J_k the n equidistant nodes

$$s_{k,p} = (p-1)\frac{2\pi}{n} \in J_k, \qquad p = 1, 2, \ldots, n.$$

The total number of nodes $s_{k,p}$ in the total parameter domain J is $(m+1)n$. We denote these $(m+1)n$ nodes by t_i, $i = 1, 2, \ldots, (m+1)n$, i.e.,

$$t_{kn+p} = s_{k,p} \in J, \qquad k = 0, 1, \ldots, m, \quad p = 1, 2, \ldots, n. \tag{3.1}$$

We define the $(m+1)n \times 1$ vector $\mathbf{t}$ by

$$\mathbf{t} = (t_1, t_2, \ldots, t_{(m+1)n})^T,$$

where T denotes transposition. For any function $\gamma(t)$ defined on J, we define $\gamma(\mathbf{t})$ as the $(m+1)n \times 1$ vector obtained by componentwise evaluation of the function $\gamma(t)$ at the points t_i, $i = 1, 2, \ldots, (m+1)n$.

ETNA
Kent State University
http://etna.math.kent.edu

As in MATLAB code, for any two vectors $\mathbf{x}$ and $\mathbf{y}$, we define $\mathbf{x}.*\mathbf{y}$ as the componentwise vector product of $\mathbf{x}$ and $\mathbf{y}$. If $\mathbf{y}_j \neq 0$ for all $j = 1, 2 \ldots, (m+1)n$, we define $\mathbf{x}./\mathbf{y}$ as the componentwise vector division of $\mathbf{x}$ by $\mathbf{y}$. For simplicity, we denote $\mathbf{x}.*\mathbf{y}$ by $\mathbf{xy}$ and $\mathbf{x}./\mathbf{y}$ by $\frac{\mathbf{x}}{\mathbf{y}}$.

Let $\gamma \in H$ be a given function. Then, in the notation of Section 2, the function γ can be written as

$$\gamma(t) = \begin{cases} \gamma_0(t), & t \in J_0, \\ \gamma_1(t), & t \in J_1, \\ \quad \vdots & \\ \gamma_m(t), & t \in J_m, \end{cases}$$

with 2π-periodic Hölder continuous real functions $\gamma_0, \gamma_1, \ldots, \gamma_m$. Thus, by the trapezoidal rule, we obtain

$$\int_J \gamma(t)dt = \sum_{k=0}^{m} \int_{J_k} \gamma_k(t)dt \approx \frac{2\pi}{n} \sum_{k=0}^{m} \sum_{p=1}^{n} \gamma_k(s_{k,p}) = \frac{2\pi}{n} \sum_{j=1}^{(m+1)n} \gamma(t_j). \tag{3.2}$$

In this paper, we use the function `zfmm2dpart` in the MATLAB toolbox FMMLIB2D developed by Greengard and Gimbutas [13] to compute complex-valued sums of the form

$$\sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{1}{\eta(t_i) - \eta(t_j)} x_j, \qquad i = 1, 2, \ldots, (m+1)n, \tag{3.3}$$

where x_j are real or complex numbers. Let $\mathbf{x}$ be the $(m+1)n \times 1$ vector

$$\mathbf{x} = (x_1, x_2, \ldots, x_{(m+1)n})^T$$

and E be the $(m+1)n \times (m+1)n$ matrix with elements

$$(E)_{ij} := \begin{cases} 0, & i = j, \\ \dfrac{1}{\eta(t_i) - \eta(t_j)}, & i \neq j, \end{cases} \quad i, j = 1, 2, \ldots, (m+1)n. \tag{3.4}$$

Then equation (3.3) can be written as a matrix-vector product $E\mathbf{x}$. Let $\mathbf{a}$ be the $2 \times (m+1)n$ real vector

$$\mathbf{a} = \begin{bmatrix} \operatorname{Re} \eta(\mathbf{t})^T \\ \operatorname{Im} \eta(\mathbf{t})^T \end{bmatrix}.$$

The matrix-vector product $E\mathbf{x}$ can be computed in $O((m+1)n)$ operations using the MATLAB function `zfmm2dpart` by

$$E\mathbf{x} = \texttt{zfmm2dpart(iprec, (m+1)n,}\mathbf{a}, \mathbf{x}^T, \texttt{1}), \tag{3.5}$$

where the tolerance of the FMM is 0.5×10^{-3} for `iprec` $= 1$, 0.5×10^{-6} for `iprec` $= 2$, 0.5×10^{-9} for `iprec` $= 3$, 0.5×10^{-12} for `iprec` $= 4$, and 0.5×10^{-15} for `iprec` $= 5$.

Similarly, the MATLAB function `zfmm2dpart` can be used to compute complex-valued sums of the form

$$\sum_{j=1}^{(m+1)n} \frac{1}{z_i - \eta(t_j)} x_j, \qquad i = 1, 2, \ldots, \hat{n}, \tag{3.6}$$

ETNA
Kent State University
http://etna.math.kent.edu

where x_j are real or complex constants and z_i are $\hat{n}$ given points in G with a given positive integer $\hat{n}$. Let F be the $\hat{n} \times (m+1)n$ matrix with elements

$$(F)_{ij} := \frac{1}{z_i - \eta(t_j)}, \qquad i = 1, 2, \ldots, \hat{n}, \quad j = 1, 2, \ldots, (m+1)n. \tag{3.7}$$

Let $\mathbf{x}$ be the $(m+1)n \times 1$ vector $\mathbf{x} = (x_1, x_2, \ldots, x_{(m+1)n})^T$, $\mathbf{z}$ be the $\hat{n} \times 1$ complex vector $\mathbf{z} = (z_1, z_2, \ldots, z_{\hat{n}})$, and $\mathbf{d}$ be the $2 \times (m+1)n$ real vector

$$\mathbf{d} = \begin{bmatrix} \operatorname{Re} \mathbf{z} \\ \operatorname{Im} \mathbf{z} \end{bmatrix}.$$

Then equation (3.6) can be written as a matrix-vector product $F\mathbf{x}$. This product can be computed using the MATLAB function `zfmm2dpart` in $O((m+1)n + \hat{n})$ operations by

$$F\mathbf{x} = \texttt{zfmm2dpart(iprec, (m+1)n,} \mathbf{a}, \mathbf{x}^T, \texttt{0, 0, 0,} \hat{n}, \mathbf{d}, \texttt{1, 0, 0)}. \tag{3.8}$$

4. Solving the integral equation with the generalized Neumann kernel.

4.1. The integral equation. We use singularity subtraction to rewrite the operators $\mathbf{N}$ and $\mathbf{M}$ to make them more suitable for using the FMM. Such a procedure is useful for solving the integral equation (2.3) for domains with corners; see [4, 22, 23, 39, 44, 47]. It is also useful for solving the integral equation (2.3) for domains with close-to-touching boundaries; see Examples 8.1 and 8.2 below.

It is known that the constant function is an eigenfunction of the generalized Neumann kernel N corresponding to the eigenvalue $\lambda = -1$ and an eigenfunction of the singular kernel M corresponding to the eigenvalue $\lambda = 0$ [31, 42], i.e.,

$$\int_J N(s,t)dt = -1, \qquad \int_J M(s,t)dt = 0.$$

Thus, the integral equation (2.3) can be written as

$$2\mu(s) - \int_J N(s,t)[\mu(t) - \mu(s)]dt = -\phi(s), \tag{4.1}$$

where

$$\phi(s) = \int_J M(s,t)[\gamma(t) - \gamma(s)]dt. \tag{4.2}$$

The integral equation (4.1) is valid even if the boundary Γ is piecewise smooth; see [44].

4.2. The Nyström method. Discretizing the integral in (4.1) by the trapezoidal rule (3.2) and substituting $s = t_i$, we obtain the linear system

$$2\mu(t_i) - \frac{2\pi}{n} \sum_{j=1}^{(m+1)n} N(t_i, t_j)[\mu(t_j) - \mu(t_i)] = -\phi(t_i), \qquad i = 1, 2, \ldots, (m+1)n.$$

Since $N(s,t)$ is continuous, the term under the summation sign is zero when $j = i$. Thus, using the notation $\mathbf{x} = \mu(\mathbf{t})$ and $\mathbf{y} = \phi(\mathbf{t})$, the linear system can be written as

$$\left(2 + \sum_{\substack{j=1 \\ j \neq i}}^{(m+1)n} \frac{2\pi}{n} N(t_i, t_j) \right) \mathbf{x}_i - \sum_{\substack{j=1 \\ j \neq i}}^{(m+1)n} \frac{2\pi}{n} N(t_i, t_j) \mathbf{x}_j = -\mathbf{y}_i, \tag{4.3}$$

ETNA
Kent State University
http://etna.math.kent.edu

for $i = 1, 2, \ldots, (m+1)n$. Let B be the $(m+1)n \times (m+1)n$ matrix with elements

$$(B)_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \dfrac{2\pi}{n} N(t_i, t_j), & \text{if } i \neq j, \quad i, j = 1, 2, \ldots, (m+1)n. \end{cases} \tag{4.4}$$

Then the $(m+1)n \times (m+1)n$ linear system (4.3) can be written as

$$(2I + \operatorname{diag}(B\mathbf{1}) - B)\mathbf{x} = -\mathbf{y}. \tag{4.5}$$

4.3. Computing the vector y. In this subsection, we present a method for computing the values of the function $\phi(t)$ defined by (4.2) at the points t_i, for $i = 1, 2, \ldots, (m+1)n$. The method can be used for all Hölder continuous functions γ without any differentiability requirement. Thus, the method presented here improves the method in [39], where the function γ was assumed to be continuously differentiable.

We rewrite the index i for $i = 1, 2, \ldots, (m+1)n$ as

$$i = kn + p,$$

where $k = 0, 1, \ldots, m$ and $p = 1, 2, \ldots, n$. Hence, by definition of the points t_i, we need to compute the values

$$y_i = \phi(t_i) = \phi(t_{kn+p}) = \phi_k(s_{k,p}). \tag{4.6}$$

By (4.2), we have

$$\phi_k(s_{k,p}) = \int_J M(s,t)[\gamma(t) - \gamma_k(s_{kp})]dt = \sum_{l=0}^{m} \int_{J_l} M(s_{k,p}, t)[\gamma_l(t) - \gamma_k(s_{k,p})]dt,$$

which, in view of (2.2), implies that

$$\begin{aligned} \phi_k(s_{k,p}) = &\int_{J_k} \frac{-1}{2\pi} \cot \frac{s_{k,p} - t}{2} [\gamma_k(t) - \gamma_k(s_{k,p})]dt \\ &+ \int_{J_k} M_1(s_{k,p}, t)[\gamma_k(t) - \gamma_k(s_{k,p})]dt \\ &+ \sum_{\substack{l=0 \\ l \neq k}}^{m} \int_{J_l} M(s_{k,p}, t)[\gamma_l(t) - \gamma_k(s_{k,p})]dt. \end{aligned} \tag{4.7}$$

The integral with the cotangent kernel in (4.7) can be discretized by Wittich's method [51]. The kernel M_1 is continuous and the kernel M is continuous for $l \neq k$. Thus, the integrals with kernels M_1 and M in (4.7) are discretized by the trapezoidal rule. Hence, we obtain

$$\begin{aligned} \phi_k(s_{kp}) = &\sum_{q=1}^{n} [-(K)_{pq}] [\gamma_k(s_{kq}) - \gamma_k(s_{kp})] \\ &+ \sum_{q=1}^{n} \frac{2\pi}{n} M_1(s_{kp}, s_{kq})[\gamma_k(s_{kq}) - \gamma_k(s_{kp})] \\ &+ \sum_{\substack{l=0 \\ l \neq k}}^{m} \sum_{q=1}^{n} \frac{2\pi}{n} M(s_{kp}, s_{lq})[\gamma_l(s_{lq}) - \gamma_k(s_{kp})], \end{aligned} \tag{4.8}$$

ETNA
Kent State University
http://etna.math.kent.edu

where the elements $(K)_{pq}$ of Wittich's matrix are given by

$$(K)_{pq} = \begin{cases} 0, & \text{if } p-q \text{ even}, \\ \frac{2}{n}\cot\frac{(p-q)\pi}{n}, & \text{if } p-q \text{ odd}, \quad p,q = 1,2,\ldots,n. \end{cases}$$

Since $(K)_{pq} = 0$ whenever $q = p$ and $M_1(s,t)$ is continuous, the term under the first two summation signs in (4.8) is zero for $q = p$. By (2.2), we have for $p \neq q$,

$$\frac{2\pi}{n}M_1(s_{kp}, s_{kq}) = \frac{1}{n}\cot\frac{s_{kp}-s_{kq}}{2} + \frac{2\pi}{n}M(s_{kp}, s_{kq}) = \frac{1}{n}\cot\frac{(p-q)\pi}{n} + \frac{2\pi}{n}M(s_{kp}, s_{kq}).$$

Hence, we obtain for $p \neq q$ that

$$\begin{aligned} -(K)_{pq} + \frac{2\pi}{n}M_1(s_{kp}, s_{kq}) &= -(K)_{pq} + \frac{1}{n}\cot\frac{(p-q)\pi}{n} + \frac{2\pi}{n}M(s_{kp}, s_{kq}) \\ &= (-1)^{p-q}\frac{1}{n}\cot\frac{(p-q)\pi}{n} + \frac{2\pi}{n}M(s_{kp}, s_{kq}). \end{aligned}$$

Let L be the $n \times n$ matrix whose elements are given by

$$(L)_{pq} = \begin{cases} 0, & \text{if } p = q, \\ (-1)^{p-q}\frac{1}{n}\cot\frac{(p-q)\pi}{n}, & \text{if } p \neq q, \quad p,q = 1,2,\ldots,n. \end{cases}$$

Thus we have

$$-(K)_{pq} + \frac{2\pi}{n}M_1(s_{kp}, s_{kq}) = (L)_{pq} + \frac{2\pi}{n}M(s_{kp}, s_{kq}), \qquad p \neq q. \tag{4.9}$$

In view of (4.9), equation (4.8) can be written as

$$\begin{aligned} \phi_k(s_{kp}) = &\sum_{q=1}^{n}(L)_{pq}[\gamma_k(s_{kq}) - \gamma_k(s_{kp})] \\ &+ \sum_{\substack{q=1 \\ q\neq p}}^{n}\frac{2\pi}{n}M(s_{kp}, s_{kq})[\gamma_k(s_{kq}) - \gamma_k(s_{kp})] \\ &+ \sum_{\substack{l=0 \\ l\neq k}}^{m}\sum_{q=1}^{n}\frac{2\pi}{n}M(s_{kp}, s_{lq})[\gamma_l(s_{lq}) - \gamma_k(s_{kp})]. \end{aligned} \tag{4.10}$$

Let D be the $(m+1)n \times (m+1)n$ matrix with elements

$$(D)_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \frac{2\pi}{n}M(t_i, t_j), & \text{if } i \neq j, \quad i,j = 1,2,\ldots,(m+1)n, \end{cases}$$

and $\hat{L}$ be the $(m+1)n \times (m+1)n$ matrix

$$\hat{L} = \begin{bmatrix} L & O & \cdots & O \\ O & L & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & L \end{bmatrix}.$$

ETNA
Kent State University
http://etna.math.kent.edu

Then, in view of (3.1), equation (4.10) reads as

$$\phi(t_i) = \sum_{j=1}^{(m+1)n} (\hat{L})_{ij}[\gamma(t_j) - \gamma(t_i)] + \sum_{j=1}^{(m+1)n} (D)_{ij}[\gamma(t_j) - \gamma(t_i)], \qquad i = 1, 2, \ldots, (m+1)n. \tag{4.11}$$

Hence, it follows from (4.6) and (4.11) that the $(m+1)n \times 1$ vector $\mathbf{y} = \phi(\mathbf{t})$ can be expressed in matrix form as

$$\mathbf{y} = D\gamma(\mathbf{t}) - \operatorname{diag}(D\mathbf{1})\gamma(\mathbf{t}) + \hat{L}\gamma(\mathbf{t}) - \operatorname{diag}(\hat{L}\mathbf{1})\gamma(\mathbf{t}).$$

For the matrix-vector product $\hat{L}\gamma(\mathbf{t})$, we have

$$\hat{L}\gamma(\mathbf{t}) = \begin{bmatrix} L & O & \cdots & O \\ O & L & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & L \end{bmatrix} \gamma(\mathbf{t}) = \begin{bmatrix} L\gamma(\mathbf{s}_1) \\ L\gamma(\mathbf{s}_2) \\ \vdots \\ L\gamma(\mathbf{s}_m) \end{bmatrix}.$$

The matrix L is circulant since it has the structure

$$L = \begin{bmatrix} b_1 & b_n & \cdots & b_3 & b_2 \\ b_2 & b_1 & \ddots & b_4 & b_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{n-1} & b_{n-2} & \ddots & b_1 & b_n \\ b_n & b_{n-1} & \cdots & b_2 & b_1 \end{bmatrix},$$

where $b_1 = 0$ and

$$b_i = (-1)^{i-1} \frac{1}{n} \cot \frac{(i-1)\pi}{n}, \qquad \text{for } i = 2, 3, \ldots, n.$$

Thus, the matrix-vector product $L\gamma(\mathbf{s}_k)$ can be computed in $O(n \ln n)$ operations using the FFT. Employing the MATLAB functions `fft` for the forward FFT and `ifft` for the inverse FFT, the vector $L\gamma(\mathbf{s}_k)$ is computed by [6, p. 92]

$$L\gamma(\mathbf{s}_k) = \texttt{ifft}\left(\texttt{fft}(\mathbf{b}) .* \texttt{fft}(\gamma(\mathbf{s}_k))\right). \tag{4.12}$$

Hence, the matrix-vector product $\hat{L}\gamma(\mathbf{t})$ can be evaluated in $O((m+1)n \ln n)$ operations.

Since the `fft` of a constant function is zero, in view of (4.12), we have

$$\hat{L}\mathbf{1} = 0.$$

Thus, the vector $\mathbf{y}$ can be written as

$$\mathbf{y} = D\gamma(\mathbf{t}) - \operatorname{diag}(D\mathbf{1})\gamma(\mathbf{t}) + \hat{L}\gamma(\mathbf{t}). \tag{4.13}$$

For the matrix-vector product $D\gamma(\mathbf{t})$, we have for $i = 1, 2, \ldots, (m+1)n$,

$$\sum_{j=1}^{(m+1)n} (D)_{ij}\gamma(t_j) = \sum_{\substack{j=1 \\ j \neq i}}^{(m+1)n} \frac{2}{n} \operatorname{Re}\left[\frac{A(t_i)}{A(t_j)} \frac{\dot{\eta}(t_j)}{\eta(t_j) - \eta(t_i)}\right] \gamma(t_j)$$

$$= -\frac{2}{n} \operatorname{Re}\left[A(t_i) \sum_{\substack{j=1 \\ j \neq i}}^{\hat{n}} \frac{1}{\eta(t_i) - \eta(t_j)} \frac{\dot{\eta}(t_j)}{A(t_j)} \gamma(t_j)\right].$$

ETNA
Kent State University
http://etna.math.kent.edu

Hence, the matrix-vector product $D\gamma(\mathbf{t})$ can be expressed in terms of the matrix E as

$$D\gamma(\mathbf{t}) = -\frac{2}{n}\,\mathrm{Re}\left[A(\mathbf{t})\left(E\left(\frac{\dot\eta(\mathbf{t})}{A(\mathbf{t})}\gamma(\mathbf{t})\right)\right)\right]. \tag{4.14}$$

It is clear from (4.14) that computing $D\gamma(\mathbf{t})$ requires one multiplication of the matrix E by a vector, which can be computed as in (3.5) by the FMM in $O((m+1)n)$ operations. The matrix-vector product $D\mathbf{1}$ can also be evaluated in $O((m+1)n)$ operations.

Hence, the vector $\mathbf{y}$ on the right-hand side of the linear system (4.5) can be computed through (4.13) in $O((m+1)n\ln n)$ operations.

4.4. Multiplication by the coefficient matrix $2I + \mathrm{diag}(B\mathbf{1}) - B$. For multiplying the matrix B by the vector $\mathbf{x}$, we have for $i = 1, 2, \ldots, (m+1)n$,

$$\begin{aligned}\sum_{j=1}^{(m+1)n}(B)_{ij}\mathbf{x}_j &= \sum_{\substack{j=1\\ j\neq i}}^{(m+1)n}\frac{2}{n}\,\mathrm{Im}\left[\frac{A(t_i)}{A(t_j)}\frac{\dot\eta(t_j)}{\eta(t_j)-\eta(t_i)}\right]\mathbf{x}_j\\ &= -\frac{2}{n}\,\mathrm{Im}\left[A(t_i)\sum_{\substack{j=1\\ j\neq i}}^{(m+1)n}\frac{1}{\eta(t_i)-\eta(t_j)}\frac{\dot\eta(t_j)}{A(t_j)}\mathbf{x}_j\right].\end{aligned}$$

Hence, the matrix-vector product $B\mathbf{x}$ can be written in terms of the matrix E as

$$B\mathbf{x} = -\frac{2}{n}\,\mathrm{Im}\left[A(\mathbf{t})\left(E\left(\frac{\dot\eta(\mathbf{t})}{A(\mathbf{t})}\mathbf{x}\right)\right)\right]. \tag{4.15}$$

It is clear from (4.15) that multiplying the matrix B by the vector $\mathbf{x}$ requires one multiplication of the matrix E by a vector, which can be computed in $O((m+1)n)$ operations. The matrix-vector product $B\mathbf{1}$ can be also computed in $O((m+1)n)$ operations. The multiplication of the diagonal matrix $2I + \mathrm{diag}(B\mathbf{1})$ by the vector $\mathbf{x}$ requires $O((m+1)n)$ operations. Thus, the multiplication of the coefficient matrix of the linear system (4.5), $2I + \mathrm{diag}(B\mathbf{1}) - B$, by the vector $\mathbf{x}$ involves $O((m+1)n)$ operations.

4.5. The MATLAB function `fbie`. We use the MATLAB function `gmres` to solve the linear system (4.5), which can be used with the matrix-vector product function $f_B(\mathbf{x})$ defined by

$$f_B(\mathbf{x}) = (2I + \mathrm{diag}(B\mathbf{1}) - B)\mathbf{x}. \tag{4.16}$$

Based on (4.15) and (3.5), the values of the function $f_B(\mathbf{x})$ can be computed by the MATLAB function `zfmm2dpart`. The linear system (4.5) can then be solved using `gmres` by

$$\mathbf{x} = \texttt{gmres}(@(\mathbf{x})f_B(\mathbf{x}), -\mathbf{y}, \texttt{restart}, \texttt{tol}, \texttt{maxit}), \tag{4.17}$$

which restarts every `restart` inner iterations, where `tol` is the tolerance of the method and `maxit` is the maximum number of outer iterations. With $\mathbf{x}$, we obtain an approximation to the solution μ of the integral equation (2.3) at the points $\mathbf{t}$ by $\mu(\mathbf{t}) = \mathbf{x}$.

In view of (4.5) and (4.13), the discretization matrices of the operators $\mathbf{I} - \mathbf{N}$ and $\mathbf{M}$ are $(2I + \mathrm{diag}(B\mathbf{1}) - B)$ and $D - \mathrm{diag}(D\mathbf{1}) + \hat L$, respectively. In view of (2.4), the $(m+1)n \times 1$ vector $h(\mathbf{t})$, which components are the values of $h(t)$ at the points $\mathbf{t}$, can be approximated by

$$h(\mathbf{t}) = \frac{1}{2}\left([D - \mathrm{diag}(D\mathbf{1}) + \hat L]\mu(\mathbf{t}) - [2 + \mathrm{diag}(B\mathbf{1}) - B]\gamma(\mathbf{t})\right). \tag{4.18}$$

ETNA
Kent State University
http://etna.math.kent.edu

Computing the vector $h(\mathbf{t})$ in [39, Eq. (62)] required computing $\dot{\mu}(\mathbf{t})$, i.e., the derivative of the solution of the integral equation (2.3) at the points $\mathbf{t}$. Thus, the method presented in (4.18) improves that in [39, Eq. (62)] since no differentiability of the function μ is required in (4.18).

Solving the linear system requires one application of the function $f_B(\mathbf{x})$ in (4.16) for each iteration of the GMRES method. Thus, each iteration of the GMRES method requires $O(m+1)n$ operations. Due to the clustering of the eigenvalues of the coefficient matrix of the linear system (see Theorem 2.3(a) and the note below Theorem 2.3), the GMRES iteration usually converges after a few steps. Even for a domain of connectivity 201 with complex geometry and close-to-touching boundaries (see Figure 8.15 below), the GMRES method required only 65 iterations to converge with accuracy 10^{-12}; see Figure 8.17 below. Thus, for sufficiently large n, the number of operation required by the GMRES method is less than the number of operations required for computing the right-hand side $\mathbf{y}$, which is $O((m+1)n\ln n)$ operations. Hence, solving the linear system (4.5) by (4.17) requires $O((m+1)n\ln n)$ operations. Similarly, computing the vector $h(\mathbf{t})$ requires $O((m+1)n\ln n)$ operations.

A MATLAB function `fbie` for the fast solution of the integral equation (2.3) and the fast computation of the function h in (2.4) using the method presented in this section is shown in Figure 4.1.

5. Solving the integral equation with the adjoint generalized Neumann kernel.

5.1. The integral equation. The integral equation (2.5) will be rewritten by applying singularity subtraction into a more suitable form for using the FMM. As for the integral equation (2.3), the singularity subtraction is useful for solving the integral equation (2.5) for domains with corners and for domains with close-to-touching boundaries. We have [53]

$$\begin{aligned} N^*(s,t) = -\tilde{N}(s,t) &= -\frac{1}{\pi}\,\mathrm{Im}\left(\frac{\tilde{A}(s)}{\tilde{A}(t)}\,\frac{\dot{\eta}(t)}{\eta(t)-\eta(s)}\right) \\ &= -\frac{1}{\pi}\,\mathrm{Im}\left(\frac{\dot{\eta}(t)}{\eta(t)-\eta(s)}\right) + \frac{1}{\pi}\,\mathrm{Im}\left(\frac{\tilde{A}(t)-\tilde{A}(s)}{\eta(t)-\eta(s)}\,\frac{\dot{\eta}(t)}{\tilde{A}(t)}\right), \end{aligned}$$

where $\tilde{A}(t) = \dot{\eta}(t)/A(t)$. Hence,

$$N^*(s,t) = -N_k(s,t) + N_g(s,t), \tag{5.1}$$

where

$$N_k(s,t) = \frac{1}{\pi}\,\mathrm{Im}\left(\frac{\dot{\eta}(t)}{\eta(t)-\eta(s)}\right) \tag{5.2}$$

is the well-known Neumann kernel and

$$N_g(s,t) = \frac{1}{\pi}\,\mathrm{Im}\left(\frac{\tilde{A}(t)-\tilde{A}(s)}{\eta(t)-\eta(s)}\,\frac{\dot{\eta}(t)}{\tilde{A}(t)}\right) = \frac{1}{\pi}\,\mathrm{Im}\left(\frac{A(s)\dot{\eta}(t)-A(t)\dot{\eta}(s)}{A(s)(\eta(t)-\eta(s))}\right).$$

The kernel $N_g(s,t)$ is continuous with

$$N_g(t,t) = \frac{1}{\pi}\,\mathrm{Im}\left(\frac{\ddot{\eta}(t)}{\dot{\eta}(t)} - \frac{\dot{A}(t)}{A(t)}\right). \tag{5.3}$$

ETNA
Kent State University
http://etna.math.kent.edu


```
function [mu,h]=fbie(et,etp,A,gam,n,iprec,restart,gmrestol,maxit)
%The function
%        [mu,h]=fbie(et,etp,A,gam,n,iprec,restart,gmrestol,maxit)
%return the unique solution mu of the integral equation
%               (I-N)mu=-Mgam
%and the function
%                h=[(I-N)gam-Mmu]/2,
%where et is the parameterization of the boundary, etp=et',
%A=exp(-i\thet)(et-alp) for bounded G and by A=exp(-i\thet)
%for unbounded G, gam is a given function, n is the number of
%nodes in each boundary component, iprec is the FMM precision
%flag, restart is the maximum number of the GMRES inner
%iterations, gmrestol is the tolerance of the GMRES method,
%and maxit is the maximum number of the GMRES outer
%iterations
a        = [real(et.') ; imag(et.')];
m        =  length(et)/n-1;
b1       = [etp./A].';
[Ub1]    = zfmm2dpart(iprec,(m+1)*n,a,b1,1);
Eone     = (Ub1.pot).';
b(1,1) = 0;
for k=2:n
    b(k,1) = (-1)^(k+1)*(1/n)*cot(pi*(k-1)/n);
end
mu     = gmres(@(x)fB(x),-fC(gam),restart,gmrestol,maxit);
h       = (fC(mu)-fB(gam))./2;
%%
function  hx  = fB (x)
    bx2   = [x.*etp./A].';
    [Ubx2]= zfmm2dpart(iprec,(m+1)*n,a,bx2,1);
    Ex    = (Ubx2.pot).';
    hx    =  2.*x-(2/n).*imag(A.*Eone).*x+(2/n).*imag(A.*Ex);
end
function  hx = fC (x)
    bx    = [x.*etp./A].';
    [Ubx] = zfmm2dpart(iprec,(m+1)*n,a,bx,1);
    Ex    = (Ubx.pot).';
    for k=1:m+1
        hLx(1+(k-1)*n:k*n,1) = ...
        ifft(fft(b).*fft(x(1+(k-1)*n:k*n,1)));
    end
    hx    = -(2/n).*real(A.*Ex)+(2/n).*real(A.*Eone).*x+hLx;
end
end
```

FIG. 4.1. *The MATLAB function* `fbie`.

The constant function is an eigenfunction of the Neumann kernel N_k corresponding to the eigenvalue $\lambda = 1$ for bounded G and to the eigenvalue $\lambda = -1$ for unbounded G [42, 44]; see also [20]. Hence,

$$\int_J N_k(s,t)dt = c\,,$$

ETNA
Kent State University
http://etna.math.kent.edu

where the constant c is defined by

$$c = \begin{cases} 1, & \text{if } G \text{ is bounded,} \\ -1, & \text{if } G \text{ is unbounded.} \end{cases}$$

Thus,

$$\int_J N^*(s,t)dt = -c + r(s),$$

where

$$r(s) = \int_J N_g(s,t)dt.$$

Hence, the integral equation (2.5) can be written as

$$[1 - c + r(s)]\mu(s) + \int_J N^*(s,t)[\mu(t) - \mu(s)]dt + \int_J \delta(s,t)\mu(t)dt = \gamma(s). \tag{5.4}$$

The integral equation (5.4) is valid even if the boundary Γ is piecewise smooth; see [44].

5.2. The Nyström method. By discretizing the integral in (5.4) by the trapezoidal rule (3.2) and substituting $s = t_i$, we obtain the linear system

$$\begin{aligned}[1 - c + r(t_i)]\mu(t_i) + \frac{2\pi}{n}\sum_{j=1}^{(m+1)n} N(t_j,t_i)[\mu(t_j) - \mu(t_i)] \\ + \frac{2\pi}{n}\sum_{j=1}^{(m+1)n} \delta(t_j,t_i)\mu(t_j) = \gamma(t_i),\end{aligned} \tag{5.5}$$

for $i = 1, 2, \ldots, (m+1)n$. Since $N(s,t)$ is continuous, the term under the summation sign is zero for $j = i$. Thus, the linear system (5.5) can be written as

$$\begin{aligned}\left(1 - c + r(t_i) - \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{2\pi}{n} N(t_j,t_i)\right)\mu(t_i) + \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{2\pi}{n} N(t_j,t_i)\mu(t_j) \\ + \frac{2\pi}{n}\sum_{j=1}^{(m+1)n} \delta(t_j,t_i)\mu(t_j) = \gamma(t_i),\end{aligned} \tag{5.6}$$

for $i = 1, 2, \ldots, (m+1)n$. Let P be the $n \times n$ matrix with elements

$$(P)_{pq} = \frac{1}{n}, \qquad p, q = 1, 2, \ldots, n,$$

$\hat{P}$ be the $(m+1)n \times (m+1)n$ matrix

$$\hat{P} = \begin{bmatrix} P & O & \cdots & O \\ O & P & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & P \end{bmatrix},$$

ETNA
Kent State University
http://etna.math.kent.edu

and $\mathbf{e}$ be the $(m+1)n \times 1$ vector with elements

$$\mathbf{e}_i = 1 - c + r(t_i) - \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{2\pi}{n} N(t_j, t_i), \qquad i = 1, 2, \ldots, (m+1)n. \tag{5.7}$$

Let $\mathbf{x} = \mu(\mathbf{t})$ and B be the matrix defined by (4.4). Then the $(m+1)n \times (m+1)n$ linear system (5.6) reads as

$$(\mathrm{diag}(\mathbf{e}) + B^T + \hat{P})\mathbf{x} = \gamma(\mathbf{t}). \tag{5.8}$$

5.3. Computing the vector e. By approximating the values of the function r at the points t_i, i.e.,

$$r(t_i) = \int_J N_g(t_i, t)dt, \qquad i = 0, 1, \ldots, (m+1)n,$$

by the trapezoidal rule (3.2), we obtain

$$r(t_i) = \frac{2\pi}{n} \sum_{j=1}^{(m+1)n} N_g(t_i, t_j) = \frac{2\pi}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} N_g(t_i, t_j) + \frac{2\pi}{n} N_g(t_i, t_i). \tag{5.9}$$

By (5.1), we have

$$\frac{2\pi}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} N_g(t_i, t_j) = \frac{2\pi}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} N_k(t_i, t_j) + \frac{2\pi}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} N(t_j, t_i). \tag{5.10}$$

Thus, it follows from (5.7), (5.9), and (5.10) that

$$\mathbf{e}_i = 1 - c + \frac{2\pi}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} N_k(t_i, t_j) + \frac{2\pi}{n} N_g(t_i, t_i), \qquad i = 0, 1, \ldots, (m+1)n. \tag{5.11}$$

For $i = 0, 1, \ldots, (m+1)n$, by (5.3) and (5.2), we have

$$\frac{2\pi}{n} N_g(t_i, t_i) = \frac{2}{n} \mathrm{Im} \left(\frac{\ddot{\eta}(t_i)}{\dot{\eta}(t_i)} - \frac{\dot{A}(t_i)}{A(t_i)} \right)$$

and

$$\begin{aligned} \frac{2\pi}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} N_k(t_i, t_j) &= \frac{2}{n} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \mathrm{Im} \left(\frac{\dot{\eta}(t_j)}{\eta(t_j) - \eta(t_i)} \right) \\ &= -\frac{2}{n} \mathrm{Im} \left[\sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{1}{\eta(t_i) - \eta(t_j)} \dot{\eta}(t_j) \right]. \end{aligned}$$

Hence, in view of (3.4) and (5.11), the vector $\mathbf{e}$ can be written as

$$\mathbf{e} = 1 - c - \frac{2}{n} \mathrm{Im} \left[E\dot{\eta}(\mathbf{t}) \right] + \frac{2}{n} \mathrm{Im} \left[\frac{\ddot{\eta}(\mathbf{t})}{\dot{\eta}(\mathbf{t})} - \frac{\dot{A}(\mathbf{t})}{A(\mathbf{t})} \right].$$

Thus, computing the vector $\mathbf{e}$ requires one multiplication of the matrix E by a vector, which can be done in $O((m+1)n)$ operations.

ETNA
Kent State University
http://etna.math.kent.edu

5.4. Multiplication by the coefficient matrix $\mathbf{diag(e)} + \boldsymbol{B^T} + \hat{\boldsymbol{P}}$. For multiplying the matrix B^T by the vector $\mathbf{x}$, we have, for $i = 1, 2, \ldots, (m+1)n$, that

$$\sum_{j=1}^{(m+1)n} (B^T)_{ij}\mathbf{x}_j = \sum_{j=1}^{(m+1)n} (B)_{ji}\mathbf{x}_j = \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{2}{n} \operatorname{Im}\left[\frac{A(t_j)}{A(t_i)} \frac{\dot{\eta}(t_i)}{\eta(t_i) - \eta(t_j)}\right] \mathbf{x}_j$$

$$= \frac{2}{n} \operatorname{Im}\left[\frac{\dot{\eta}(t_i)}{A(t_i)} \sum_{\substack{j=1 \\ j\neq i}}^{(m+1)n} \frac{1}{\eta(t_i) - \eta(t_j)} A(t_j)\mathbf{x}_j\right].$$

Hence, the matrix-vector product $B^T\mathbf{x}$ can be written in terms of the matrix E as

$$B^T\mathbf{x} = \frac{2}{n} \operatorname{Im}\left[\frac{\dot{\eta}(\mathbf{t})}{A(\mathbf{t})} \left(E\left(A(\mathbf{t})\mathbf{x}\right)\right)\right]. \tag{5.12}$$

Thus, multiplying the matrix B^T by the vector $\mathbf{x}$ requires one multiplication of the matrix E by a vector, which can be computed in $O((m+1)n)$ operations.

For the matrix-vector product $\hat{P}\mathbf{x}$, we rewrite the vector $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{bmatrix},$$

where each of the vectors $\mathbf{x}_j$, $j = 0, 1, \ldots, m$, is an $n \times 1$ vector. Then the matrix-vector product $\hat{P}\mathbf{x}$ can be computed by

$$\hat{P}\mathbf{x} = \begin{bmatrix} P\mathbf{x}_0 \\ P\mathbf{x}_1 \\ \vdots \\ P\mathbf{x}_m \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\sum_{p=1}^{n} (\mathbf{x}_0)_p \\ \frac{1}{n}\sum_{p=1}^{n} (\mathbf{x}_1)_p \\ \vdots \\ \frac{1}{n}\sum_{p=1}^{n} (\mathbf{x}_m)_p \end{bmatrix}.$$

Thus, the multiplication of the matrix $\hat{P}$ by the vector $\mathbf{x}$ can be computed in $O((m+1)n)$ operations. The multiplication of the diagonal matrix $\operatorname{diag}(\mathbf{e})$ by the vector $\mathbf{x}$ can also be computed in $O((m+1)n)$ operations. Hence, the multiplication of the coefficient matrix $\operatorname{diag}(\mathbf{e}) + B^T + \hat{P}$ of the linear system (5.8) by the vector $\mathbf{x}$ requires $O((m+1)n)$ operations.

5.5. The MATLAB function `fbiead`. The linear system (5.8) will be solved using the MATLAB function `gmres` with the matrix-vector product function

$$g_B(\mathbf{x}) = (\operatorname{diag}(\mathbf{e}) + B^T + \hat{P})\mathbf{x}. \tag{5.13}$$

Based on (5.12) and (3.5), the values of the function $g_B(\mathbf{x})$ can be computed using the MATLAB function `zfmm2dpart`. The linear system (5.8) can then be solved using `gmres` by

$$\mathbf{x} = \texttt{gmres}(@(\mathbf{x})g_B(\mathbf{x}), \gamma(\mathbf{t}), \texttt{restart}, \texttt{tol}, \texttt{maxit}). \tag{5.14}$$

With $\mathbf{x}$, we obtain an approximation to the solution μ of the integral equation (2.5) at the points $\mathbf{t}$ by $\mu(\mathbf{t}) = \mathbf{x}$.

ETNA
Kent State University
http://etna.math.kent.edu

In contrast to the integral equation with the generalized Neumann kernel (2.3), the right-hand side of the integral equation with the adjoint generalized Neumann kernel (2.5) is given explicitly. Since computing the values of the function $g_B(\mathbf{x})$ in (5.13) requires $O((m+1)n)$ operations, each iteration of the GMRES method requires $O((m+1)n)$ operations. In view of Theorem 2.3(b) and the note below Theorem 2.3, the eigenvalues of the coefficient matrix of the linear system cluster around 1. Thus, GMRES usually converges after only a small number of iterations especially for domains with well-separated boundaries. Hence, solving the linear system (5.8) by (5.14) requires $O((m+1)n)$ operations. For domains with close-to-touching boundaries, the constant in the expression $O((m+1)n)$ can be large.

A MATLAB function `fbiead` for the fast solution of the integral equation (2.5) using the method presented in this section is shown in Figure 5.1.

```
function mu=fbiead(et,etp,etpp,A,Ap,gam,n,c,iprec,...
                   restart,gmrestol,maxit)
%The function
%        mu=fbiead(et,etp,etpp,A,Ap,gam,n,c,iprec,...
                   restart,gmrestol,maxit)
%returns the unique solution mu of the integral equation
%               (I+N*+J)mu=gam
%where et is the parameterization of the boundary,
%etp=et', etpp=et'', A=exp(-i\thet)(et-alp) for
%bounded G and A=exp(-i\thet) for unbounded G,
%gam is a given function, n is the number of nodes
%in each boundary component, c=1 for bounded G and
%c=-1 for unbounded G, iprec is the FMM precision flag,
%restart is the maximum number of the GMRES
%inner iterations, gmrestol is the tolerance of the
%GMRES method, and maxit is the maximum number of
%GMRES outer iterations
a      = [real(et.') ; imag(et.')];
m      =  length(et)/n-1;
[Uetp] = zfmm2dpart(iprec,(m+1)*n,a,etp.',1);
Eetp   = (Uetp.pot).';
e      = 1-c-(2/n).*imag(Eetp)+(2/n).*imag(etpp./etp-Ap./A);
mu     = gmres(@(x)gB(x),gam,restart,gmrestol,maxit);
%%
function  hx  = gB (x)
    for k=1:m+1
        hPx(1+(k-1)*n:k*n,1) = (1/n)*sum(x(1+(k-1)*n:k*n,1));
    end
    [UAx]  = zfmm2dpart(iprec,(m+1)*n,a,[A.*x].',1);
    EAx    = (UAx.pot).';
    Btx    = (2/n).*imag((etp./A).*EAx);
    hx     =  e.*x+Btx+hPx;
end
end
```

FIG. 5.1. *The MATLAB function* `fbiead`.

6. Computing the Cauchy integral formula. The solutions of the boundary integral equations (2.3) and (2.5) provide us with the values of the conformal mapping and the solution of the boundary value problem on the boundary Γ. Computing these functions at interior

ETNA
Kent State University
http://etna.math.kent.edu

points $z \in G$ requires the Cauchy integral formula

$$f(z) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(\eta)}{\eta - z} d\eta, \qquad z \in G, \tag{6.1a}$$

for bounded G and

$$f(z) = f(\infty) + \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(\eta)}{\eta - z} d\eta, \qquad z \in G, \tag{6.1b}$$

for unbounded G, where f is an analytic function on G with known boundary values on Γ. The integral in (6.1) can be approximated by the trapezoidal rule (3.2). However, the integrand in (6.1) becomes nearly singular for points $z \in G$ which are close to the boundary Γ. In this section, we review an accurate and fast numerical method for computing the Cauchy integral formulas (6.1a) and (6.1b) proposed in [5, 18, 39].

Suppose that $f(z)$ is analytic in a domain $\hat{G}$ containing $G \cup \Gamma$. Thus, the integrand in (6.1), i.e.,

$$\frac{f(\eta)}{\eta - z}$$

has a pole at $\eta = z \in \hat{G}$. For $z \in G$, we have

$$\frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{1}{\eta - z} d\eta = 1$$

for bounded G and

$$\frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{1}{\eta - z} d\eta = 0$$

for unbounded G. Thus, the Cauchy integral formula (6.1) can then be written for $z \in G$ as

$$\frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(\eta) - f(z)}{\eta - z} d\eta = 0 \tag{6.2a}$$

for bounded G and

$$f(z) = f(\infty) + \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(\eta) - f(z)}{\eta - z} d\eta \tag{6.2b}$$

for unbounded G. Thus, $\eta = z$ is not a pole of the integrand in the new formula (6.2) since

$$\frac{f(\eta) - z}{\eta - z}$$

is an analytic function of $\eta \in \hat{G}$. Hence, accurate results can be obtained if the trapezoidal rule is used to discretize the integral in (6.2); see [5] for more details.

By discretizing the integral in (6.2) using the trapezoidal rule (3.2), we obtain for $z \in G$,

$$\frac{2\pi}{n} \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{(m+1)n} \frac{f(\eta(t_j)) - f(z)}{\eta(t_j) - z} \dot{\eta}(t_j) \approx 0$$

ETNA
Kent State University
http://etna.math.kent.edu

for bounded G and

$$f(z) \approx f(\infty) + \frac{2\pi}{n}\frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{(m+1)n} \frac{f(\eta(t_j)) - f(z)}{\eta(t_j) - z}\dot{\eta}(t_j)$$

for unbounded G. Consequently, the values of the function $f(z)$ for $z \in G$ are given by

$$f(z) \approx \frac{\displaystyle\sum_{j=1}^{(m+1)n} \frac{f(\eta(t_j))\dot{\eta}(t_j)}{\eta(t_j) - z}}{\displaystyle\sum_{j=1}^{(m+1)n} \frac{\dot{\eta}(t_j)}{\eta(t_j) - z}}$$

for bounded G and

$$f(z) \approx \frac{f(\infty) + \dfrac{1}{n\mathrm{i}} \displaystyle\sum_{j=1}^{(m+1)n} \frac{f(\eta(t_j))\dot{\eta}(t_j)}{\eta(t_j) - z}}{1 + \dfrac{1}{n\mathrm{i}} \displaystyle\sum_{j=1}^{(m+1)n} \frac{\dot{\eta}(t_j)}{\eta(t_j) - z}}$$

for unbounded G.

For a given positive integer $\hat{n}$, let $z_1, z_2, \ldots, z_{\hat{n}}$ be given points in G and $\mathbf{z}$ be the $\hat{n} \times 1$ complex vector $\mathbf{z} = (z_1, z_2, \ldots, z_{\hat{n}})$. Then the values $f(z_i)$, for $i = 1, 2, \ldots, \hat{n}$, can be computed from

$$f(z_i) \approx \frac{\displaystyle\sum_{j=1}^{(m+1)n} \frac{1}{z_i - \eta(t_j)} f(\eta(t_j))\dot{\eta}(t_j)}{\displaystyle\sum_{j=1}^{(m+1)n} \frac{1}{z_i - \eta(t_j)}\dot{\eta}(t_j)} \tag{6.3a}$$

for bounded G and

$$f(z_i) \approx \frac{f(\infty) - \dfrac{1}{n\mathrm{i}} \displaystyle\sum_{j=1}^{(m+1)n} \frac{1}{z_i - \eta(t_j)} f(\eta(t_j))\dot{\eta}(t_j)}{1 - \dfrac{1}{n\mathrm{i}} \displaystyle\sum_{j=1}^{(m+1)n} \frac{1}{z_i - \eta(t_j)}\dot{\eta}(t_j)} \tag{6.3b}$$

for unbounded G. Using the matrix F defined by (3.7), the summations in (6.3) can be written as a matrix-vector product

$$f(\mathbf{z}) \approx \frac{F[f(\eta(\mathbf{t}))\dot{\eta}(\mathbf{t})]}{F[\dot{\eta}(\mathbf{t})]} \tag{6.4a}$$

for bounded G and

$$f(\mathbf{z}) \approx \frac{f(\infty) - \dfrac{1}{n\mathrm{i}} F[f(\eta(\mathbf{t}))\dot{\eta}(\mathbf{t})]}{1 - \dfrac{1}{n\mathrm{i}} F[\dot{\eta}(\mathbf{t})]} \tag{6.4b}$$

ETNA
Kent State University
http://etna.math.kent.edu


```
function  fz  = fcau (et,etp,f,z,n,finf)
%The function
%        fz  = fcau (et,etp,f,z,n,finf)
%returns the values of the analytic function f computed
%using the Cauchy integral formula at interior vector of
%points z, where et is the parameterization of the boundary,
%etp=et', finf is the values of f at  infinity for
%unbounded G, n is the number of nodes in each boundary component.
vz    = [real(z) ; imag(z)];       % target
nz    = length(z);                 % ntarget
a     = [real(et.') ; imag(et.')]; % source
tn    = length(et);                % nsource=(m+1)n
iprec = 4;                         %- FMM precision flag
bf    = [f.*etp].';
[Uf]  = zfmm2dpart(iprec,tn,a,bf,0,0,0,nz,vz,1,0,0);
b1    = [etp].';
[U1]  = zfmm2dpart(iprec,tn,a,b1,0,0,0,nz,vz,1,0,0);
if( nargin == 4 )
    fz    = (Uf.pottarg)./(U1.pottarg);
end
if( nargin == 6 )
    fz= (finf-(Uf.pottarg)./(n*i))./(1-(U1.pottarg)./(n*i));
end
end
```

FIG. 6.1. *The MATLAB function* `fcau`.

for unbounded G. Thus, computing the vector $f(\mathbf{z})$ in (6.4) requires two multiplications of the matrix F by a vector, which can be done as in (3.8) by the FMM with operations of the order of $O((m+1)n+\hat{n})$.

A MATLAB function `fcau` for the fast computation of $f(\mathbf{z})$ using the above described method is presented in Figure 6.1.

7. Domains with piecewise smooth boundary. Suppose that $\gamma(t)$ is smooth in each interval J_j except at $p_j \geq 1$ points

$$c_{j,k} = (k-1)\frac{2\pi}{p_j} \in J_j, \qquad k = 1, 2, \ldots, p_j, \quad j = 0, 1, \ldots, m.$$

Suppose that $\omega(t)$ is the bijective, strictly monotonically increasing and infinitely differentiable function defined by [22]

$$\omega(t) = 2\pi \frac{[v(t)]^p}{[v(t)]^p + [v(2\pi - t)]^p},$$

where

$$v(t) = \left(\frac{1}{p} - \frac{1}{2}\right)\left(\frac{\pi - t}{\pi}\right)^3 + \frac{1}{p}\frac{t-\pi}{\pi} + \frac{1}{2}, \qquad t \in [0, 2\pi].$$

The grading parameter p is an integer such that $p \geq 2$.

We define a function $\delta_j(t)$,

$$\delta_j(t) \ : \ [0, 2\pi] \to [0, 2\pi],$$

ETNA
Kent State University
http://etna.math.kent.edu

by

$$\delta_j(t) = \frac{1}{p_j}\omega(p_j\,(t - c_{j,k})) + c_{j,k}, \qquad t \in [c_{j,k}, c_{j,k+1}]\,,$$

for $k = 1, 2, \ldots, p_j$ and $j = 0, 1, \ldots, m$. Then the function δ_j satisfies

$$\begin{aligned} \dot{\delta}_j\,(c_{j,k}) &= 0, \qquad k = 1, 2, \ldots, p_j, \\ \dot{\delta}_j(t) &\neq 0, \qquad \text{for all } \; t \in J_j - \{c_{j,1}, c_{j,2}, \ldots, c_{j,p_j}\}. \end{aligned}$$

Thus, to compute the integral $\int_j \gamma(t)dt$, we introduce the substitution $t = \delta(\tau)$ to obtain

$$\int_J \gamma(t)dt = \int_J \gamma(\delta(\tau))\dot{\delta}(\tau)d\tau = \int_J \hat{\gamma}(\tau)d\tau\,,$$

where

$$\hat{\gamma}(\tau) = \gamma(\delta(\tau))\dot{\delta}(\tau).$$

The function $\hat{\gamma}$ is smooth on J and satisfies $\hat{\gamma}(0) = \hat{\gamma}(2\pi) = 0$. Hence, applying the trapezoidal rule (3.2) to the transformed integral yields

$$\int_J \gamma(t)dt \approx \frac{2\pi}{n}\sum_{k=0}^{m}\sum_{p=1}^{n}\hat{\gamma}(s_{k,p}) = \frac{2\pi}{n}\sum_{j=1}^{(m+1)n}\hat{\gamma}(t_j) = \frac{2\pi}{n}\sum_{j=1}^{(m+1)n}\gamma(\delta(t_j))\dot{\delta}(t_j). \tag{7.1}$$

Now, suppose that each boundary component Γ_j contains p_j corner points located at $\eta_j(c_{j,k})$, $k = 1, 2, \ldots, p_j$, $j = 0, 1, \ldots, m$. Then the integral equations (4.1) and (5.4) can be solved accurately by discretizing the integrals in (4.1) and (5.4) by the trapezoidal rule (7.1); see [22, 44, 55] for more details.

An equivalent method for the discretization of (4.1) and (5.4) is to choose a piecewise smooth parametrization $\zeta_j(t)$ of the boundary component Γ_j and then defining a parametrization $\eta_j(t)$ of Γ_j by

$$\eta_j(t) = \zeta_j(\delta_j(t)), \qquad j = 0, 1, \ldots, m.$$

The integrals in (4.1) and (5.4) can then be discretized accurately by the trapezoidal rule (3.2). See [10, 55] for more details.

8. Numerical examples. To test the performance of the functions `fbie` and `fbiead`, five numerical examples are presented. In the first and second example, we consider a bounded circular domain of connectivity 5 with variable distance ε between the boundaries. In the first example, we study the accuracy of the functions `fbie` and `fbiead` for domains whose boundaries are very close to each other. In the second example, we test the performance of the functions `fbie` and `fbiead` by computing the conformal mapping from bounded domains with close-to-touching boundaries onto the unit disc with circular slits and the unit disc with both circular and radial slits. In the third example, we present an application of the function `fbie` to solve the Dirichlet boundary value problem in a bounded multiply connected domain of very high connectivity. In the fourth and fifth example, we consider an unbounded domain appearing in a real world problem. The domain has a high connectivity and its boundaries have a complex geometry and are very close to each other. The function `fbie` is used in the fourth example to compute the complex potential of the uniform flow past islands and in the

ETNA
Kent State University
http://etna.math.kent.edu

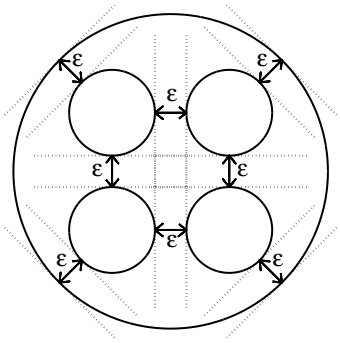

FIG. 8.1. *The domain of Examples 8.1 and 8.2.*

fifth example to compute the conformal mapping from the domain onto the unbounded circular domain.

EXAMPLE 8.1. We consider a bounded multiply connected domain G of connectivity 5 ($m = 4$). The boundary of G consists of 5 circles with a variable distance ε between them; see Figure 8.1. The external circle is the unit circle. The internal four circles have the same radius

$$\frac{2 - \varepsilon(2 + 2\sqrt{2})}{2 + 2\sqrt{2}}$$

and the centres

$$\pm\left(\frac{2-\varepsilon}{2+2\sqrt{2}}\right) \pm \mathrm{i}\left(\frac{2-\varepsilon}{2+2\sqrt{2}}\right).$$

In this example, we study the effect of (a) the distance ε, (b) the function θ, and (c) the singularity subtraction in (4.1) and (5.4) on the accuracy of the functions `fbie` and `fbiead`.

We choose the function γ in (2.3) such that the exact solution μ of the integral equation (2.3) is known. We assume that $\alpha = 0 \in G$ and $f(z)$ is defined by

$$f(z) = \sin z + \frac{1}{z-2}. \tag{8.1}$$

The function $f(z)$ is an analytic function in G. For this function f, we assume that γ is given by

$$\gamma(\eta) = \mathrm{Re}[A(\eta) f(\eta)], \tag{8.2}$$

where A is defined by (1.2). Concerning θ in (1.4), we test two cases. In the first case, we consider the constant function

$$\theta(\eta) = \pi/2, \qquad \text{for all } \eta \in \Gamma,$$

i.e., θ has the same value on all boundary components. In the second case, we consider the non-constant function

$$\theta(\eta) = (\pi/2, 0, \pi/2, 0, \pi/2).$$

Then the exact solution of the integral equation (2.3) is [31, 53]

$$\mu(\eta) = \mathrm{Im}[A(\eta) f(\eta)], \qquad \eta \in \Gamma,$$

ETNA
Kent State University
http://etna.math.kent.edu

for f given by (8.1). The approximate solution obtained using the function `fbie` is denoted by μ_n.

We also consider the numerical solution of the uniquely solvable integral equation with the adjoint generalized Neumann kernel (2.5) with γ given by (8.2). We do not have the exact solution of (2.5) available for such a function γ. However, its unique solution ϕ satisfies [42]

$$\mathbf{J}\phi = \mathbf{J}\gamma.$$

Let ϕ_n be the approximation obtained using the function `fbiead`. Then,

$$\int_J [\phi_n(t) - \gamma(t)]dt \approx \frac{2\pi}{n} \sum_{j=1}^{(m+1)n} [\phi_n(t_j) - \gamma(t_j)] \approx 0.$$

We define the error in ϕ_n by

$$E_n = \left| \frac{2\pi}{n} \sum_{j=1}^{(m+1)n} [\phi_n(t_j) - \gamma(t_j)] \right|.$$

The approximate solutions μ_n and ϕ_n are computed with `iprec` $= 4$, `restart` $= 25$, `gmrestol` $= 10^{-12}$, and `maxit` $= 40$. The numerical results are displayed in Figures 8.2–8.10. The maximum error norm $\|\mu - \mu_n\|_\infty$ for the function `fbie` and the error E_n for the function `fbiead` versus the number of nodes n in the discretization of each boundary component are displayed in Figures 8.2(a,e) for constant θ and in Figures 8.2(b,f) for non-constant θ for $\varepsilon = 0.5, 0.1, 0.001$. To show the effect of the singularity subtraction in (4.1) and (5.4) on the accuracy of the functions `fbie` and `fbiead`, we present the maximum error norm $\|\mu - \mu_n\|_\infty$ and the error E_n obtained by solving the integral equations (2.3) and (2.5) using the same method used in the functions `fbie` and `fbiead` but without singularity subtraction for the constant function θ in Figures 8.2(c,g) and for non-constant θ in Figures 8.2(d,h). The maximum error norm $\|\mu - \mu_n\|_\infty$ for `fbie` and the error E_n for `fbiead` versus the separation distance ε are displayed in Figure 8.3 (left) for constant θ and in Figure 8.3 (right) for non-constant θ. The number of GMRES iterations are shown in Figures 8.4 and 8.5. The number of GMRES iterations depends on the eigenvalues of the coefficient matrices of the linear systems, which are plotted in Figures 8.6 and 8.8 for the functions `fbie` and `fbiead`, respectively, for $\varepsilon = 0.5, 0.1, 0.001$ and $n = 1024$ (the size of the matrices is 5120×5120). For $\varepsilon = 0.001$ and $n = 4094$ (the size of the matrices is 20480×20480), the eigenvalues are displayed in Figures 8.7 and 8.9. It is clear that the eigenvalues are strongly clustered around 1 for $\varepsilon = 0.5$, i.e., the boundaries are well-separated. The eigenvalues are real for constant θ and, in general, complex for non-constant θ. The eigenvalues are in the closed disk centred at 1 with radius 1. However, for very small ε, we need to use a sufficiently large n to ensure that the eigenvalues are in the closed disk; see Figures 8.6(e,f), 8.7, 8.8(e,f), and 8.9. The condition number of the coefficient matrices of the linear systems for the functions `fbie` and `fbiead` is presented in Figure 8.10 (left). The condition number is computed using the MATLAB condition number estimation function `condest`.

ETNA
Kent State University
http://etna.math.kent.edu

We conclude the following observations from Figures 8.2–8.10:

1. For the integral equation with the generalized Neumann kernel (2.3), the singularity subtraction in (4.1) improves the accuracy of the approximate solutions only for the case when θ is a constant function. Thus, the accuracy of `fbie` is affected by θ being constant or not. We get accurate results for constant functions θ even for very small distance ε when n is sufficiently large. For a non-constant function θ, we get accurate results for moderately small ε, but then the accuracy is getting worse as the distance ε becomes very small.
2. For the integral equation with the adjoint generalized Neumann kernel (2.3), the singularity subtraction in (5.4) improves the accuracy of the approximate solution for both cases of θ. The function `fbiead` gives accurate results even for very small ε when n is sufficiently large for both cases of θ.
3. The eigenvalues of the coefficient matrices of the linear systems for both functions `fbie` and `fbiead` are strongly clustered around 1 for well-separated boundaries. For sufficiently large n, the eigenvalues are in the interval $(0, 2]$ for constant θ and in the closed disk centred at 1 with radius 1 for non-constant θ.
4. For both functions `fbie` and `fbiead`, the condition number of the coefficient matrices of the linear systems does not depend on θ being constant or not. The condition number increases as ε decreases.
5. For both functions `fbie` and `fbiead`, the number of GMRES iterations does not depend on θ being constant or not. For well-separated boundaries, only a few GMRES iterations are required for convergence. The number of GMRES iterations increases as ε decreases.

The above conclusions are confirmed by further numerical examples (not reported in this paper).

EXAMPLE 8.2. In this example, we present an application of the functions `fbie` and `fbiead` to compute the conformal mappings from the bounded multiply connected domain G of Example 8.1 (see Figure 8.1) onto the disc with circular slits and the disc with both circular and radial slits. This example shows that these functions can be used to compute the conformal mapping even for close-to-touching boundaries. For the function `fbie`, we use the methods presented in [32, 35], which are based on the integral equation with the generalized Neumann kernel (2.3). For the function `fbiead`, we use the method presented in [43, 55, 56], which is based on the integral equation with the adjoint generalized Neumann kernel (2.5). For both functions `fbie` and `fbiead`, we set `iprec` $= 4$, `restart` $= 25$, `gmrestol` $= 10^{-12}$, and `maxit` $= 40$. For the disc with circular slits, the function θ is a constant function, where $\theta(\eta) = (\pi/2, \pi/2, \pi/2, \pi/2, \pi/2)$. For the disc with both circular and radial slits, the function θ is not constant, and its values are $\pi/2$ on the external boundary, $\pi/2$ on the boundaries which is mapped to circular slits, and 0 on the boundaries which is mapped to radial slits. In this example, we assume that $\theta(\eta) = (\pi/2, 0, \pi/2, 0, \pi/2)$.

The original domains G for the separation distances $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ are displayed in Figure 8.11. The images of the original domains obtained with `fbie` for $n = 4096$ are displayed in the second row of Figure 8.11 for constant θ and in the fourth row of Figure 8.11 for non-constant θ. The images of the original domains obtained with `fbiead` for $n = 4096$ are displayed in the third row of Figure 8.11 for constant θ and in the fifth row of Figure 8.11 for non-constant θ.

As explained in Example 8.1, the function `fbie` gives accurate results for the case of a constant function θ. For the non-constant case, we get accurate results for $\varepsilon = 10^{-1}, 10^{-2}$. For $\varepsilon = 10^{-3}$ the radial slits goes outside of the unit disc, and for $\varepsilon = 10^{-4}$ the obtained figure is incorrect; see the third and fourth figures in the fourth row in Figure 8.11. The function

ETNA
Kent State University
http://etna.math.kent.edu

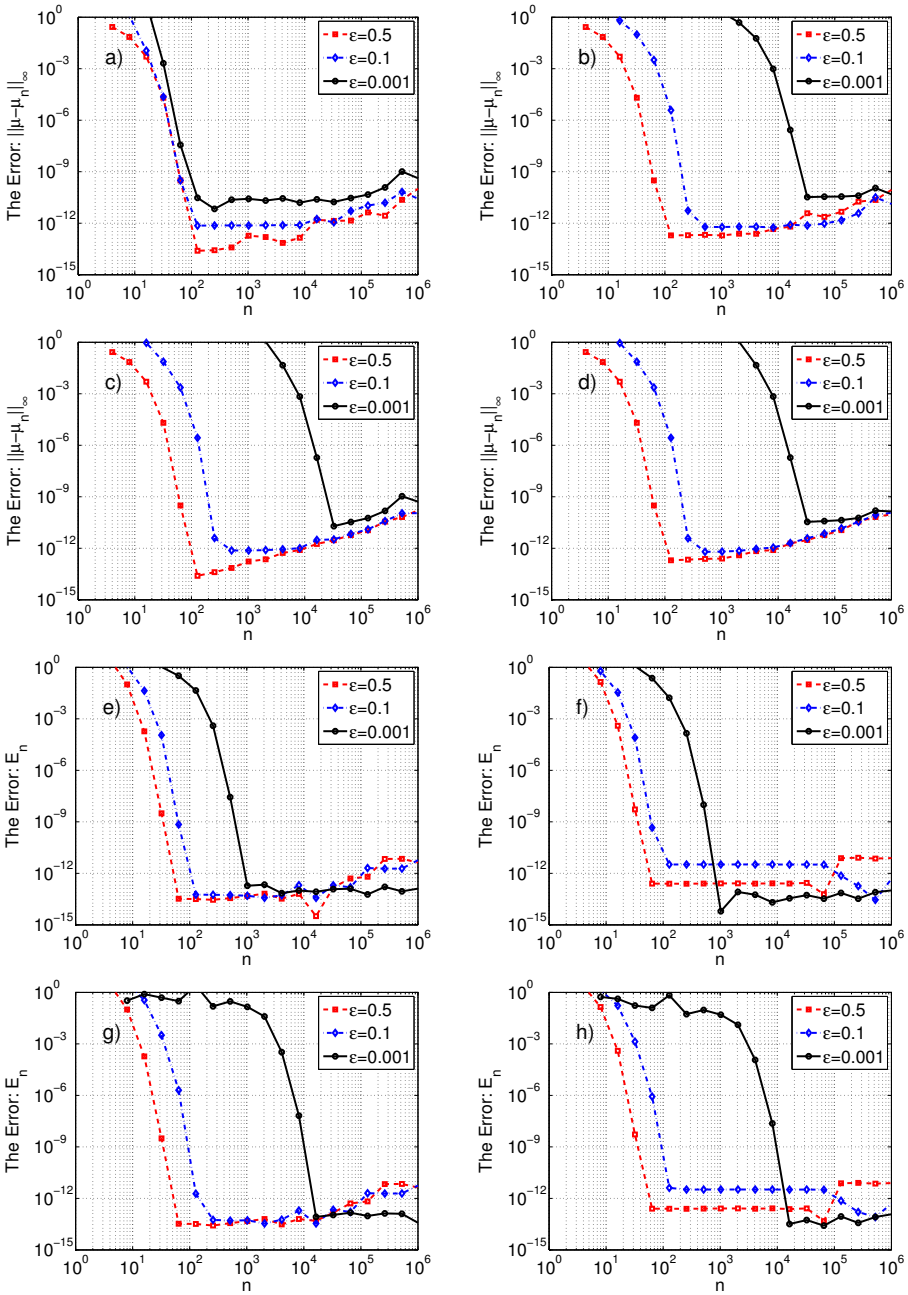

FIG. 8.2. *The maximum error norm $\|\mu - \mu_n\|_\infty$ and the error E_n versus the number of nodes n for $\varepsilon = 0.5, 0.1, 0.001$. The figures in the left column are obtained with the constant function θ, and the figures in the right column are obtained with the non-constant function θ using: (a,b)* `fbie`*, (c,d) the integral equation (2.3) without singularity subtraction, (e,d)* `fbiead`*, (g,h) the integral equation (2.5) without singularity subtraction.*

`fbiead` gives accurate results for both the constant and non-constant case of θ. However, on the one hand, the methods presented in [32, 35] are based on solving the integral equation (2.3) and then computing the function h by (2.4) to obtain the boundary values of the mapping

ETNA
Kent State University
http://etna.math.kent.edu

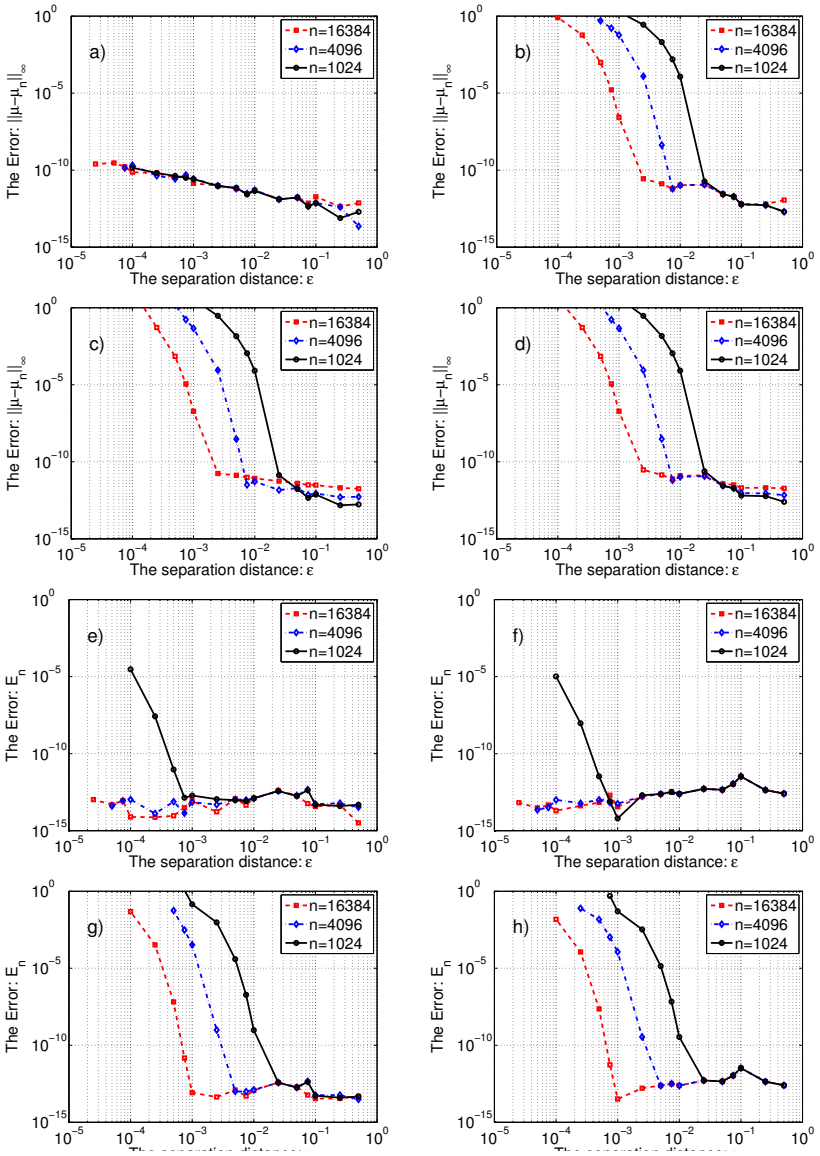

FIG. 8.3. *The maximum error norm $\|\mu - \mu_n\|_\infty$ and the error E_n as a function of the separation distance ε. The figures in the left column are obtained with the constant function θ and the figures in the right column are obtained with the non-constant function θ using: (a,b)* `fbie`*, (c,d) the integral equation (2.3) without singularity subtraction, (e,d)* `fbiead`*, (g,h) the integral equation (2.5) without singularity subtraction.*

function and the parameters of the canonical domain. Thus, the complexity of the method based on the integral equation with the generalized Neumann kernel (2.3) is $O((m+1)n \ln n)$; see [39]. On the other hand, for the method presented in [43, 55, 56], we need to solve $m+1$ integral equations to obtain the parameters of the canonical domain and one integral equation to obtain the derivative of the boundary correspondence function. The complexity of solving these $m+2$ integral equations using the function `fbiead` is $O((m+2)(m+1)n)$. To obtain the derivative of the boundary correspondence function, we need to use the FFT

ETNA
Kent State University
http://etna.math.kent.edu

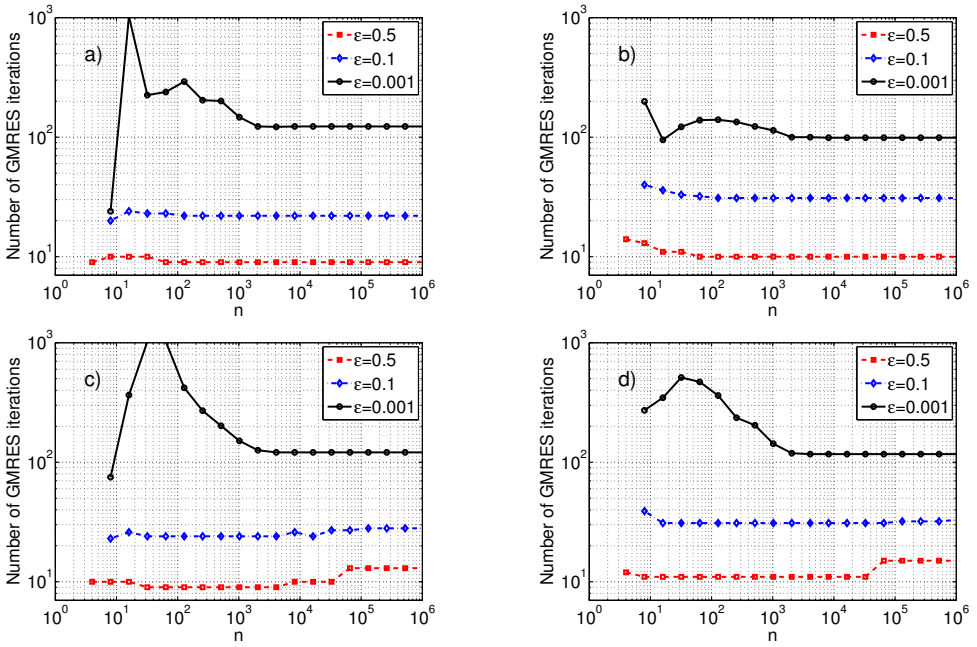

FIG. 8.4. *The total number of GMRES iterations versus the number of nodes in the discretization of each boundary component n for $\varepsilon = 0.5, 0.1, 0.001$. The figures in the left column are obtained with the constant function θ and the figures in the right column are obtained with the non-constant function θ using: (a,b)* `fbie`*, (c,d)* `fbiead`.

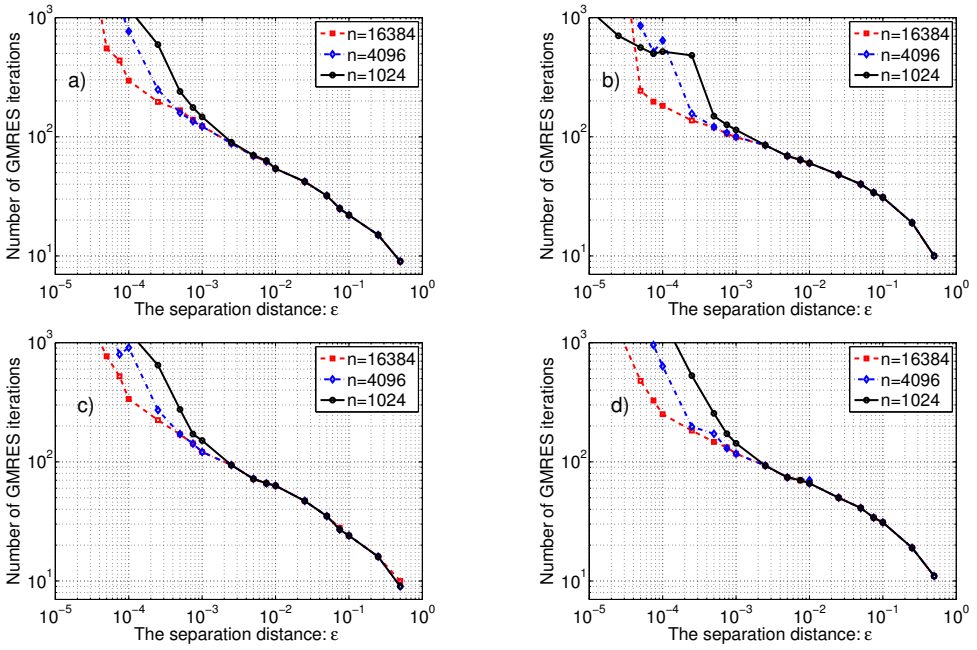

FIG. 8.5. *The total number of GMRES iterations as a function of the separation distance ε. The figures in the left column are obtained with the constant function θ and the figures in the right column are obtained with the non-constant function θ for: (a,b)* `fbie`*, (c,d)* `fbiead`.

ETNA
Kent State University
http://etna.math.kent.edu

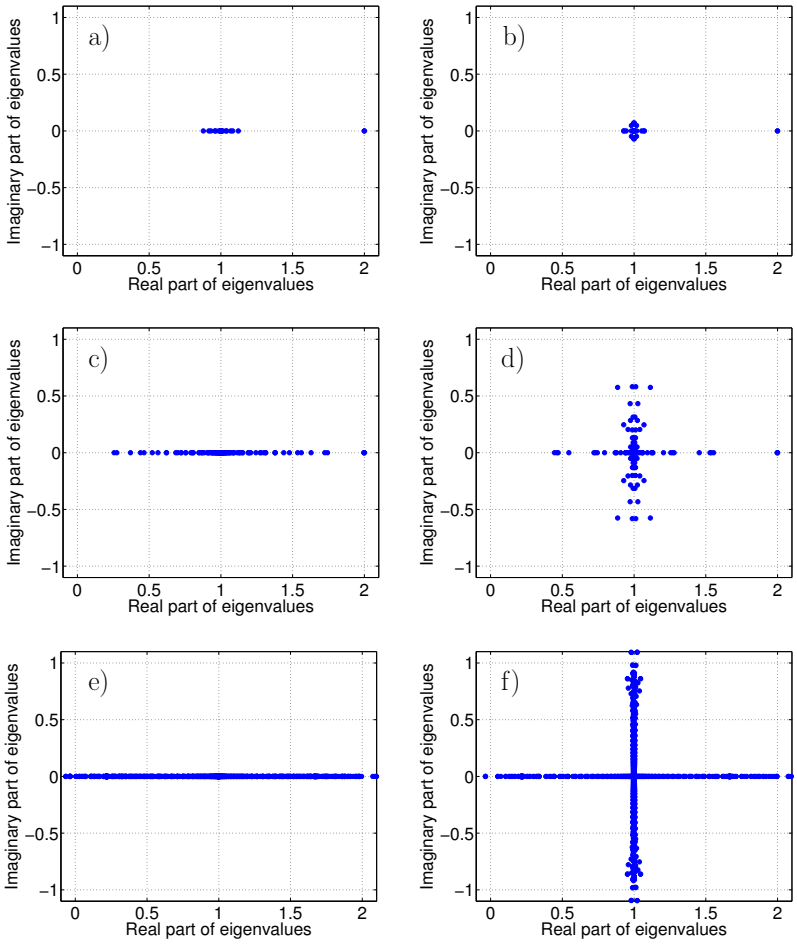

FIG. 8.6. *The eigenvalues of the coefficient matrices of the linear systems for the function* `fbie` *obtained with* $n = 1024$ *for constant* θ *(left) and non-constant* θ *(right) for: (a,b)* $\varepsilon = 0.5$*, (c,d)* $\varepsilon = 0.1$*; (e,f)* $\varepsilon = 0.001$.

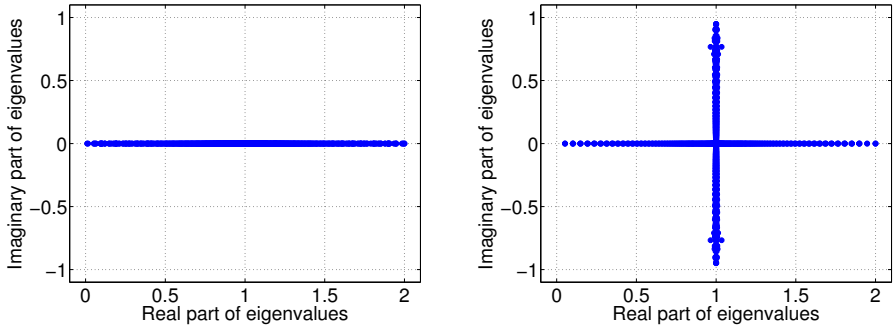

FIG. 8.7. *The eigenvalues of the coefficient matrices of the linear systems for the function* `fbie` *obtained with* $n = 4096$ *for constant* θ *(left) and non-constant* θ *(right) for* $\varepsilon = 0.001$.

ETNA
Kent State University
http://etna.math.kent.edu

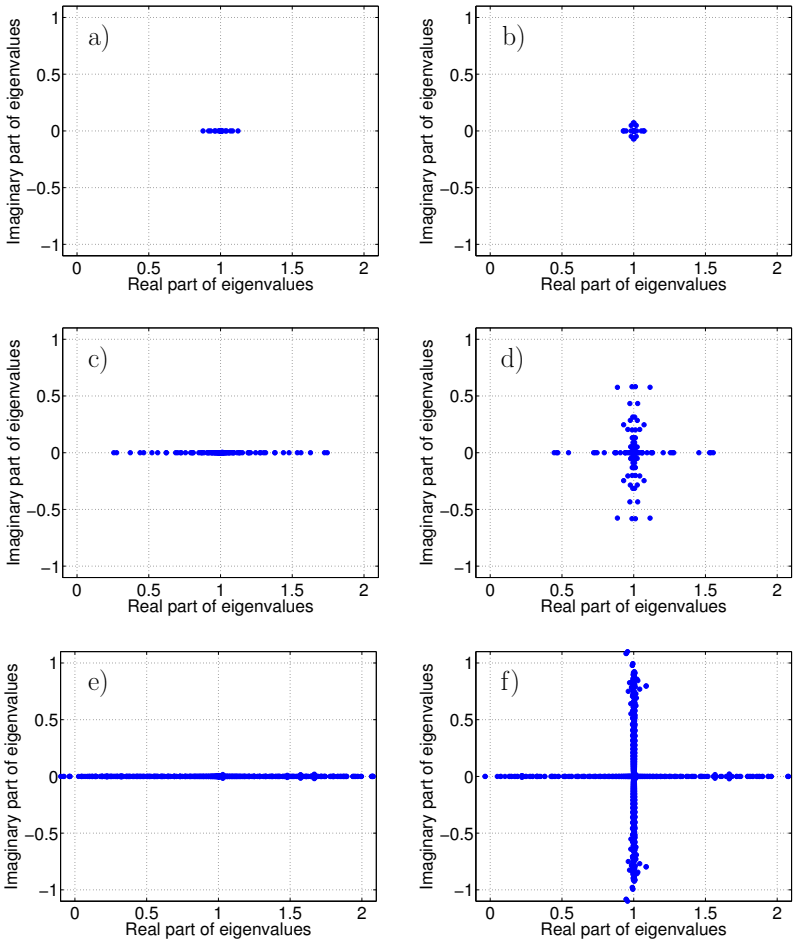

FIG. 8.8. *The eigenvalues of the coefficient matrices of the linear systems for the function* `fbiead` *obtained with* $n = 1024$ *for constant* θ *(left) and non-constant* θ *(right) for: (a,b)* $\varepsilon = 0.5$*, (c,d)* $\varepsilon = 0.1$*, (e,f)* $\varepsilon = 0.001$*.*

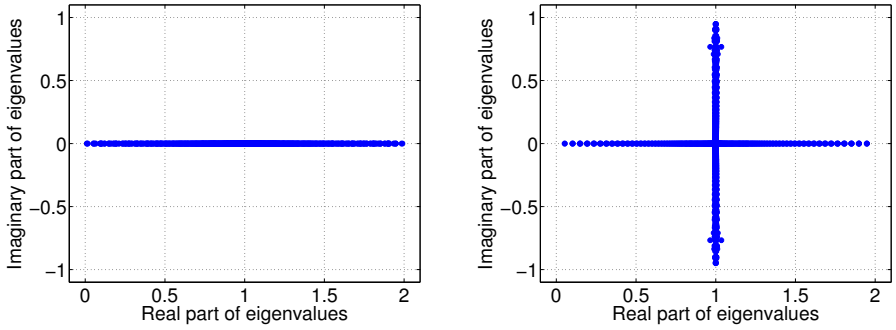

FIG. 8.9. *The eigenvalues of the coefficient matrices of the linear systems for the function* `fbiead` *obtained with* $n = 4096$ *for constant* θ *(left) and non-constant* θ *(right) for* $\varepsilon = 0.001$*.*

ETNA
Kent State University
http://etna.math.kent.edu

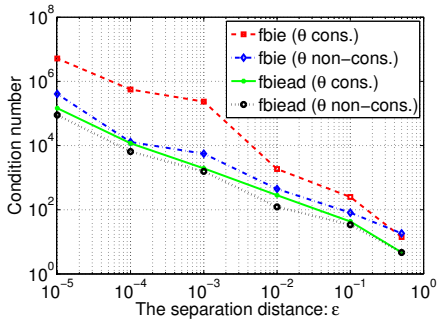

FIG. 8.10. *The condition number of the coefficient matrices of the linear systems as a function of the separation distance ε obtained with $n = 1024$.*

for each boundary component J_j, $j = 0, 1, \ldots, m$, to compute the boundary values of the mapping function. The complexity of computing the boundary correspondence function from its derivative is $O((m+1)n \ln n)$. Thus, the complexity of the method based on the integral equation with the adjoint generalized Neumann kernel (2.5) is $O((m+1)(m+2+\ln n)n)$. Thus, as Figure 8.12 indicates, the total number of GMRES iterations as well as the CPU time of the method based on the integral equation with the adjoint generalized Neumann kernel (2.5) is much larger than the number of GMRES iterations and the CPU time of the method based on the integral equation with the generalized Neumann kernel (2.3).

EXAMPLE 8.3. We consider a bounded multiply connected domain G of connectivity 1089 ($m = 1088$); see Figure 8.13. The boundary consists of 544 circles and 545 squares including the external boundary. We assume that $\alpha = 0 \in G$. We consider an application of the function `fbie` to solve the Laplace equation with Dirichlet boundary condition, which requires determining a real function u such that

$$\nabla^2 u = 0, \qquad \text{in } G, \tag{8.3a}$$

$$u = \gamma, \qquad \text{on } \Gamma, \tag{8.3b}$$

where γ is a given continuous function on Γ.

The Dirichlet problem (8.3) has a unique solution u [11, 30]. The function u is the real part of a function $F(z)$ which is analytic in the domain G but is not necessary single-valued. However, the function $F(z)$ can be written as

$$F(z) = g(z) - \sum_{j=1}^{m} a_j \log(z - z_j)\,,$$

where g is a single-valued analytic function in G, each z_j is a given point inside Γ_j for $j = 1, 2, \ldots, m$, and $a_1, a_2, \ldots, a_m$ are unknown real constants. The constants a_j are chosen to ensure that $g(z)$ is single-valued. These constants are uniquely determined by γ. In particular, if γ is the real part of a single-valued analytic function in G, then $a_j = 0$ for all j. For more details, see [11, 27, 30, 42].

Since we are interested only in computing the real part $u(z) = \operatorname{Re} F(z)$, we may assume that $c = g(\alpha)$ is real. Hence, the function $f(z)$ defined by

$$f(z) = \frac{g(z) - c}{z - \alpha}$$

ETNA
Kent State University
http://etna.math.kent.edu

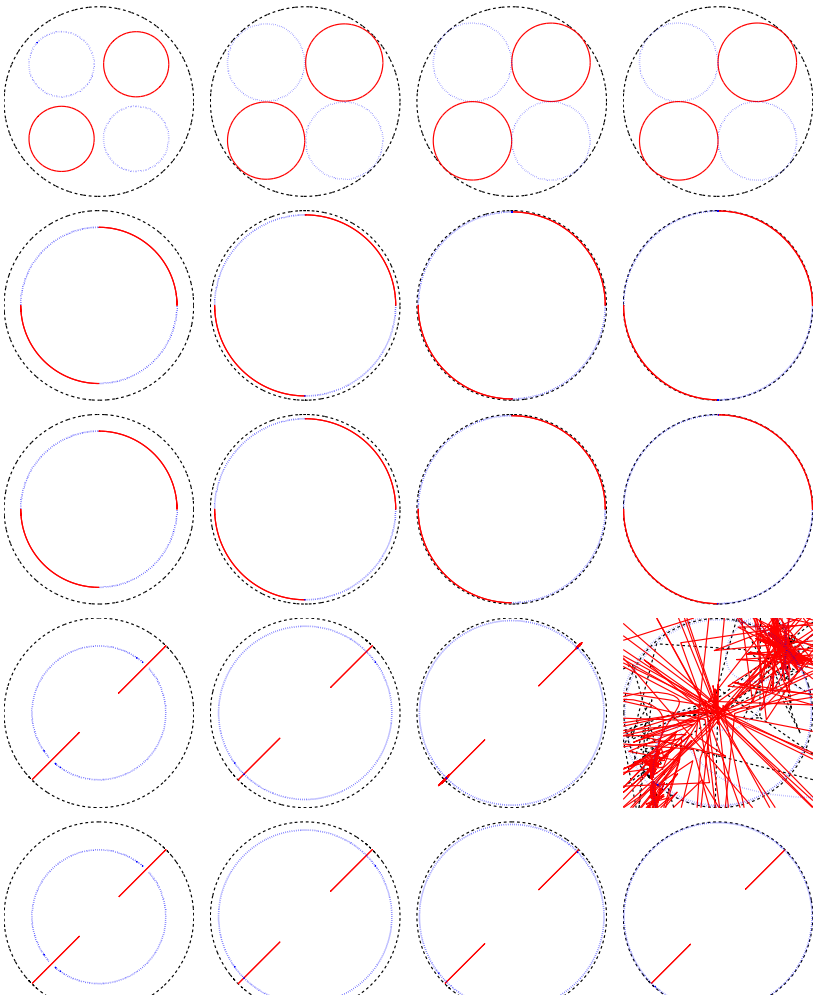

FIG. 8.11. *First row: the domains G for the separation distances $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. Second row: the images of G obtained with* `fbie` *for the constant function θ. Third row: the images of G obtained with* `fbiead` *for the constant function θ. Fourth row: the images of G obtained with* `fbie` *for the non-constant function θ. Fifth row: the images of G obtained with* `fbiead` *for the non-constant function θ.*

is analytic in G. Thus, $F(z)$ can be written as

$$F(z) = (z-\alpha)f(z) + c - \sum_{j=1}^{m} a_j \log(z - z_j). \tag{8.4}$$

Since $\mathrm{Re}[F] = \gamma$ on the boundary Γ, it follows from (8.4) that the function f is the unique solution of the Riemann-Hilbert problem

$$\mathrm{Re}[A(\eta)f(\eta)] = \gamma(\eta) - c + \sum_{j=1}^{m} a_j \log|\eta - z_j|, \qquad \eta \in \Gamma, \tag{8.5}$$

where the function A is given by (1.2) with $\theta_j = \frac{\pi}{2}$, for all $j = 0, 1, \ldots, m$. The unknown

ETNA
Kent State University
http://etna.math.kent.edu

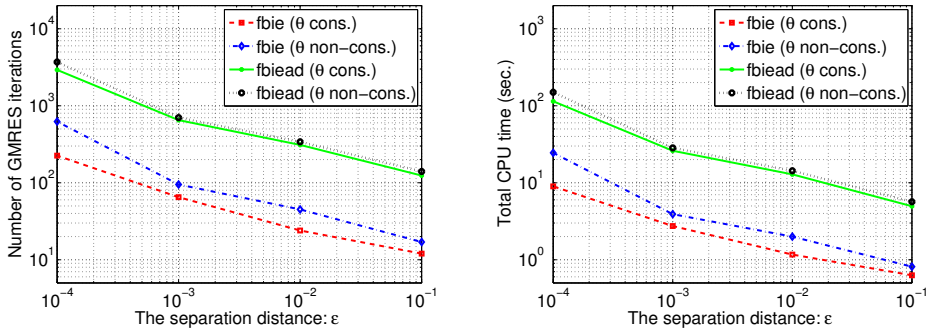

FIG. 8.12. *The total number of GMRES iterations and the total CPU time as a function of the separation distance ε obtained with $n = 4096$.*

$m + 1$ real constants $c, a_1, \ldots, a_m$ on the right-hand side of (8.5) are chosen to ensure that the Riemann-Hilbert problem (8.5) is solvable.

The boundary integral equation (2.3) has been used in [42] to compute the analytic function $f(z)$ as well as the constants $c, a_1, \ldots, a_m$ and hence the functions $g(z)$ and $F(z)$. The unique solution u is then given by $u = \operatorname{Re} F$. For more details, we refer the reader to [42].

Here we chose γ in (8.3b) such that the unique solution u of the Dirichlet problem (8.3) is the real part of

$$F(z) = \sin z + \frac{2}{z-2}, \qquad z \in G.$$

Since $F(z)$ is a single-valued analytic function in G, it follows that $a_j = 0$ for all j. Thus,

$$F(z) = g(z) = (z-\alpha)f(z) + c\,,$$

where $\alpha = 0$ and $c = g(\alpha) = F(\alpha) = -1$. The function

$$f(z) = \frac{1}{z}\left(\sin z + \frac{2}{z-2} + 1\right)$$

is the unique solution of the Riemann-Hilbert problem (1.5) with

$$\gamma(\eta) = \operatorname{Re}\left[\sin\eta + (\eta-2)^{-1}\right], \qquad \eta \in \Gamma, \tag{8.6}$$

and

$$h(\eta) = -c = 1, \qquad \eta \in \Gamma. \tag{8.7}$$

For the function γ in (8.6), we use `fbie` to solve the integral equation (2.3) and to compute h in (2.4). For such γ, the exact solution of the integral equation (2.3) is

$$\mu(\eta) = \operatorname{Im}[A(\eta)f(\eta(\eta))] = \operatorname{Im}\left[\sin\eta + (\eta-2)^{-1}\right], \qquad \eta \in \Gamma,$$

and the exact value of h in (2.4) is given by (8.7).

Suppose that μ_n and h_n are the approximate solutions obtained using `fbie`. The values of the maximum error norms $\|\mu - \mu_n\|_\infty$ and $\|h - h_n\|_\infty$ versus the total number of nodes are displayed in Figure 8.14. Figure 8.14 also exhibits the total CPU time (in seconds) and the number of GMRES iterations required to obtain the approximate solutions μ_n, h_n using `fbie` versus the total number of nodes. Finally, Figure 8.14 displays the relative residual

ETNA
Kent State University
http://etna.math.kent.edu

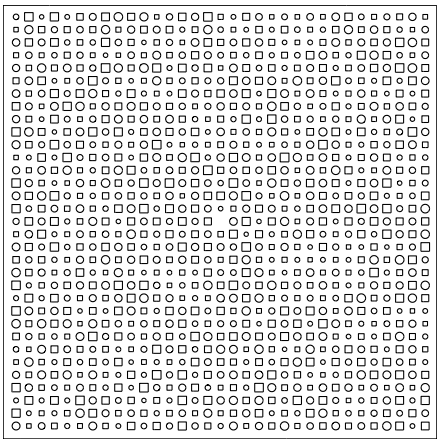

FIG. 8.13. *The domain of Example 8.3.*

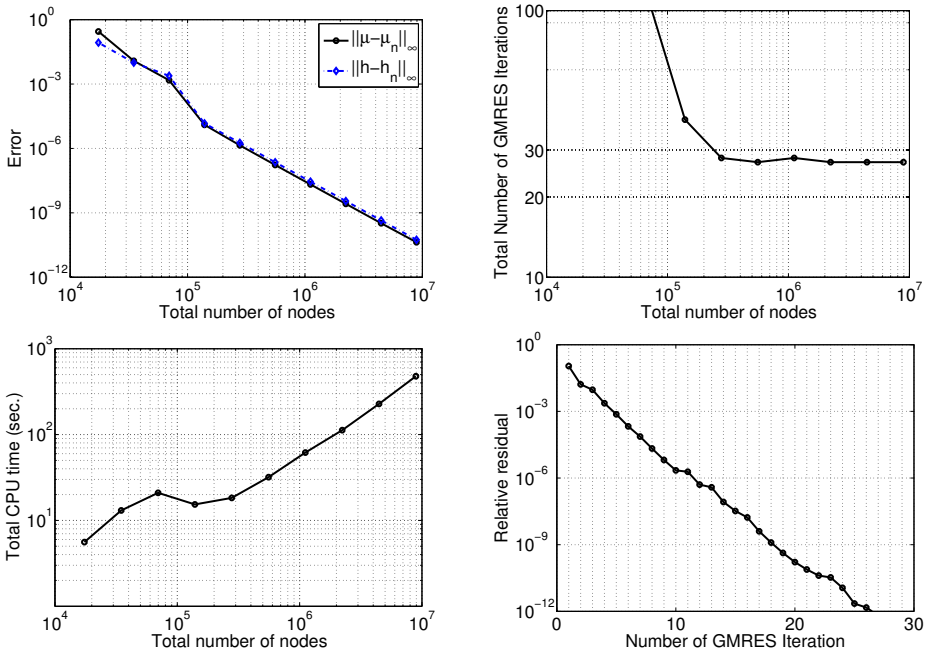

FIG. 8.14. *Numerical results for Example 8.3.*

versus the number of iterations of GMRES obtained with $n = 8192$ (the total number of nodes is 8921088). The numerical results are obtained with `iprec` $= 4$, `restart` $= 10$, `gmrestol` $= 10^{-12}$, and `maxit` $= 10$. For the piecewise smooth boundaries, we use the method describe in Section 7 with grading parameter $p = 3$.

EXAMPLE 8.4. In this example, we present an application of the described method to a real world problem. We consider the unbounded domain G of connectivity 210 (i.e., $m = 209$), exterior to an artificial archipelago located in the waters of the Arabian Gulf, 4 kilometres off the coast of Dubai, and known as "The World Islands". An aerial image of "The World Islands" is shown in Figure 8.15 (left). The boundaries of the islands extracted from the aerial

ETNA
Kent State University
http://etna.math.kent.edu

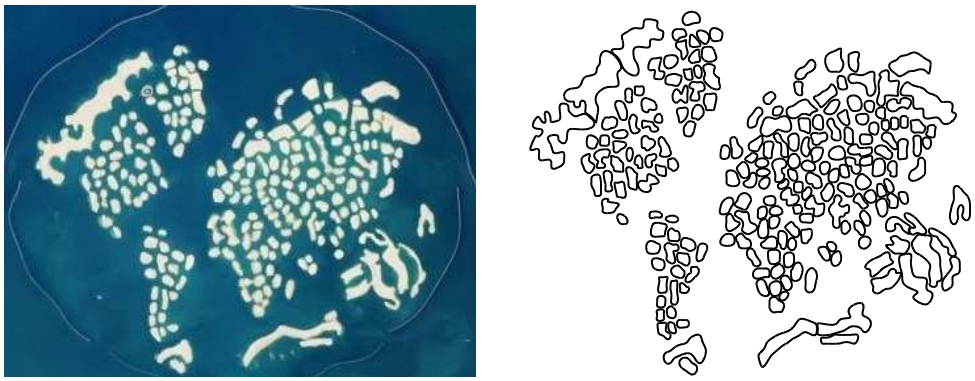

FIG. 8.15. *The domain for Examples 8.4 and 8.5. An aerial photograph of "The World Islands" (left) and the boundaries of the islands extracted from the image (right).*

image are displayed in Figure 8.15 (right). The boundaries are parameterized by trigonometric interpolating polynomials. It is clear from Figure 8.15 (right) that the boundaries are very close to each other, but they do not touch each other.

In this example, we use the integral equation with the generalized Neumann kernel to find solutions for a steady irrotational uniform flow past the islands in Figure 8.15. Suppose that $F(z)$ is the complex potential of the flow where $z \in G \cup \Gamma$. We assume that the flow speed at infinity is U and makes an angle ν with the positive real axis. We also assume that χ_j is the circulation of the fluid along each boundary component Γ_j for $j = 0, 1, \ldots, m-1$. Then the complex potential $F(z)$ can be written as (see [27, p. 158] and [34])

$$F(z) = Ue^{-\mathrm{i}\nu}z + \sum_{j=0}^{m} \frac{\chi_j}{2\pi\mathrm{i}} \log(z - z_j) + f(z) + c\,,$$

where $f(z)$ is a single-valued analytic function in G with $f(\infty) = 0$, c is a complex constant, and z_j denotes an arbitrary given points inside Γ_j. The constant c has no effect on the velocity field, so we assume that $c = 0$. Hence,

$$\mathrm{i}^2 f(z) = -F(z) + Ue^{-\mathrm{i}\nu}z + \sum_{j=0}^{m} \frac{\chi_j}{2\pi\mathrm{i}} \log(z - z_j), \qquad z \in G \cup \Gamma. \tag{8.8}$$

The complex potential $F(z)$ must satisfy the boundary condition that $\operatorname{Im} F(z)$ is constant on each boundary component Γ_j so that the boundaries are streamlines (see [26, p. 180], [27, p. 158] and [8]), i.e.,

$$\operatorname{Im}[F(\eta)] = c_j, \quad \eta \in \Gamma_j, \qquad j = 0, 1, \ldots, m, \tag{8.9}$$

with real constants $c_0, c_1, \ldots, c_m$. We assume that the function A is given by (1.2) with $\theta_j = 0$ for all $j = 0, 1, \ldots, m$, i.e., $A(\eta) = \mathrm{i}$ for all $\eta \in \Gamma$. Then, by taking the imaginary part on both sides of (8.8) and using (8.9), we conclude that $f(z)$ is a solution of the Riemann-Hilbert problem (1.5) with

$$h = (-c_0, -c_1, \ldots, -c_m) \tag{8.10}$$

and

$$\gamma(\eta) = U \operatorname{Im}[e^{-\mathrm{i}\nu}\eta] - \sum_{j=0}^{m} \frac{\chi_j}{2\pi} \log|z - z_j|.$$

ETNA
Kent State University
http://etna.math.kent.edu

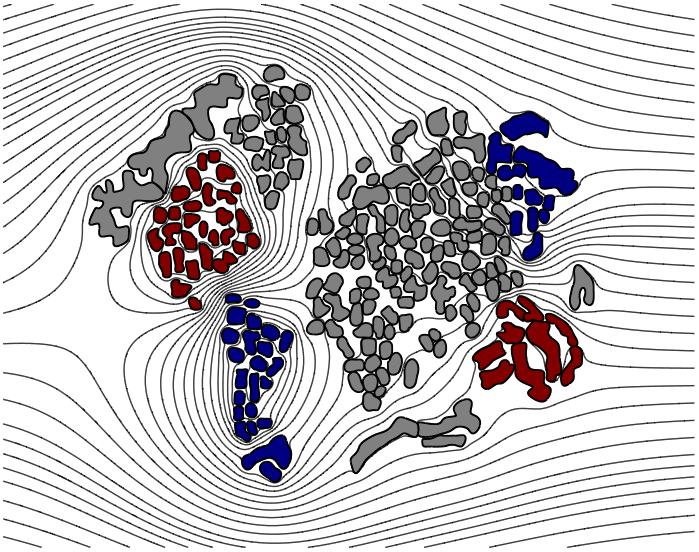

FIG. 8.16. *The streamlines for uniform flow past the* 210 *islands for* $U = 1$ *and* $\nu = 0$. *The circulation around the islands filled with red colour is* -1, *the circulation around the islands filled with blue colour is* $+1$, *and the circulation around the rest of the islands is* 0.

We use the function `fbie` to obtain an approximate solution μ_n to the integral equation (2.3) and an approximation h_n to the function h. Then approximations to the boundary values of the function f are given by $f_n = (\gamma + h_n + \mathrm{i}\mu_n)/A$. The values of the approximate function $f_n(z)$ for $z \in G$ are calculated by the Cauchy integral formula. Hence, an approximation of the complex potential $F(z)$ is given by

$$F_n(z) = Ue^{-\mathrm{i}\nu}z + \sum_{j=0}^{m} \frac{\chi_j}{2\pi\mathrm{i}} \log(z - z_j) + f_n(z), \qquad z \in G \cup \Gamma.$$

If one is interested in computing the values of the constants c_j in (8.9), then, in view of (8.10), we can compute these from the approximate function h_n.

Figure 8.16 displays the streamlines for uniform flow past the 210 islands. In the function `fbie`, we choose the parameters `iprec` $= 4$, `restart` $= 25$, `gmrestol` $= 10^{-12}$, `maxit` $= 10$. We use $n = 8192$ nodes in the discretization of each boundary component, so the total number of nodes is 1720320. The total number of GMRES iterations versus n is displayed in Figure 8.17. For $n = 8192$, the GMRES method converges after around 65 iterations, which is acceptable for such complicated domains. For such cases, a possible way to reduce the number of iterations is to use preconditioning techniques, which are not considered in this paper; see, e.g., [1, 6, 12, 19, 25].

EXAMPLE 8.5. In this example, we compute the conformal mapping from the unbounded multiply connected domain G of Example 8.4 (see Figure 8.15) onto the unbounded multiply connected circular domain, i.e., a domain all of whose boundaries are circles.

Recently, a fast numerical implementation of Koebe's iterative method for computing the conformal mapping onto circular domains has been presented in [38]. The method is based on the integral equation with the generalized Neumann kernel (2.3) and can be used to compute the conformal mapping, its derivative, and its inverse. The integral equation was solved in [38] by the MATLAB function `fbie` presented in this paper. Thus, the numerical results presented

ETNA
Kent State University
http://etna.math.kent.edu

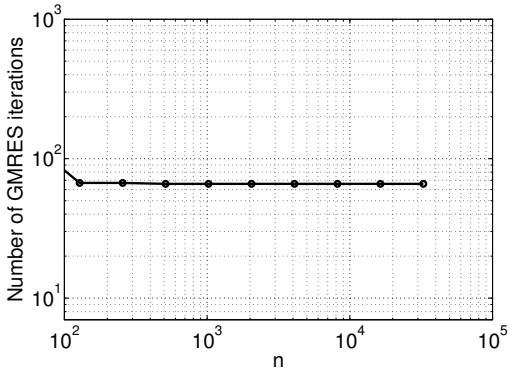

FIG. 8.17. *The total number of GMRES iterations versus n.*

in [38] provide extra numerical verification of the function `fbie`. For more details, we refer the reader to [38].

In this example, as an application of the function `fbie`, we use the method presented in [38] to compute the conformal mapping from the unbounded multiply connected domain G in Figure 8.15 onto the multiply connected circular domain. This example shows that the function `fbie` can be used to compute conformal mappings even for domains with complex and close-to-touching boundaries. The numerical results obtained with $n = 16384$, `iprec` $= 4$, `restart` $= 25$, `gmrestol` $= 10^{-12}$, and `maxit` $= 10$ are displayed in Figure 8.18.

9. Conclusions. In this paper, two new numerical methods for a fast computation of the solutions of the boundary integral equations with the generalized Neumann kernel and the adjoint generalized Neumann kernel are presented. The methods are based on discretizing the integral equations using the Nyström method with the trapezoidal rule and then solving the obtained linear systems using the GMRES method combined with the FMM. The complexity of the presented methods is $O((m+1)n \ln n)$ for the integral equations with the generalized Neumann kernel and $O((m+1)n)$ for the integral equations with the adjoint generalized Neumann kernel. The described methods are fast, accurate, and can be used for domains with close-to-touching boundaries, domains of very high connectivity, domains with piecewise smooth boundaries, and domains appearing in real world problems.

Furthermore, two MATLAB functions `fbie` and `fbiead` are presented for a fast computation of the solutions of the integral equations with the generalized Neumann kernel and the adjoint generalized Neumann kernel, respectively. The functions `fbie` and `fbiead` have been validated on five numerical examples for various types of multiply connected domains.

As listed in Table 1.1, the integral equations with these kernels have applications to several problems in applied mathematics. Thus, the functions `fbie` and `fbiead` will be useful for researchers interested in such problems. These functions can also be used for solving several other tasks related to fluid dynamics and conformal mappings. So far, the function `fbie` has been used in [38] to develop a fast method for computing the conformal mapping onto circular domains, in [45] to develop a fast computational method for potential flows in multiply connected coastal domains, and in [37] to develop a fast method for computing the conformal mapping between simply connected domains. The numerical results presented in [37, 38, 45] provide extra verifications of the usefulness of the function `fbie`.

Acknowledgement. The author would like to thank the anonymous referees for their valuable comments and suggestions which improved the presentation of this paper. The author

ETNA
Kent State University
http://etna.math.kent.edu

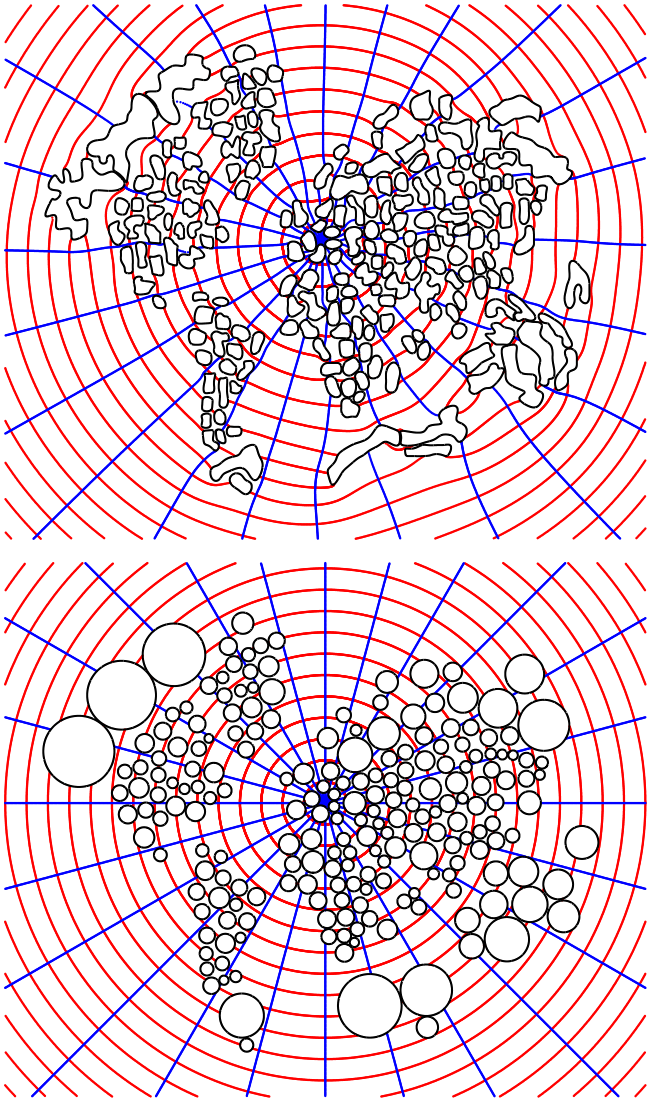

FIG. 8.18. *The original domain G (above) and its circular image (below). The figures show orthogonal polar grids drawn in the circular domain (below) and their pre-images in the original domain G (above). The pre-images have been computed by the inverse of the conformal mapping.*

also thanks Prof. Leslie Greengard and Dr. Zydrunas Gimbutas for making the MATLAB toolbox FMMLIB2D [13] publicly available.

REFERENCES

[1] A. S. AL-FHAID, S. SERRA-CAPIZZANO, D. SESANA, AND M. Z. ULLAH, *Singular-value (and eigenvalue) distribution and Krylov preconditioning of sequences of sampling matrices approximating integral operators*, Numer. Linear Algebra Appl., 21 (2014), pp. 722–743.

[2] S. AL-HATEMI, A. MURID, AND M. NASSER, *Solving a mixed boundary value problem via an integral equation with adjoint generalized Neumann kernel in bounded multiply connected regions*, in Proceedings of the 20th National Symposium on Mathematical Sciences, Malaysia, A. Ishak, I. Hashim , E. S. Ismail, and R. Nazar, eds., AIP Conf. Proc., 1522, AIP Publishing, Melville, 2013, pp. 508–517.

ETNA
Kent State University
http://etna.math.kent.edu

[3] ———, *A boundary integral equation with the generalized Neumann kernel for a mixed boundary value problem in unbounded multiply connected regions*, Bound. Value Probl., 2013:54 (2013), (17 pages).
[4] P. M. ANSELONE, *Singularity subtraction in the numerical solution of integral equations*, J. Austral. Math. Soc. Ser. B, 22 (1980/81), pp. 408–418.
[5] A. P. AUSTIN, P. KRAVANJA, AND L. N. TREFETHEN, *Numerical algorithms based on analytic function values at roots of unity*, SIAM J. Numer. Anal., 52 (2014), pp. 1795–1821.
[6] K. CHEN, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, Cambridge, 2005.
[7] D. CROWDY, *Analytical solutions for uniform potential flow past multiple cylinders*, Eur. J. Mech. B Fluids, 25 (2006), pp. 459–470.
[8] ———, *Calculating the lift on a finite stack of cylindrical aerofoils*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 462 (2006), pp. 1387–1407.
[9] ———, *The Schwarz problem in multiply connected domains and the Schottky-Klein prime function*, Complex Var. Elliptic Equ., 53 (2008), pp. 221–236.
[10] J. ELSCHNER AND I. G. GRAHAM, *Quadrature methods for Symm's integral equation on polygons*, IMA J. Numer. Anal., 17 (1997), pp. 643–664.
[11] F. D. GAKHOV, *Boundary Value Problems*, Pergamon Press, Oxford, 1966.
[12] A. GREENBAUM, L. GREENGARD, AND G. B. MCFADDEN, *Laplace's equation and the Dirichlet-Neumann map in multiply connected domains*, J. Comput. Phys., 105 (1993), pp. 267–278.
[13] L. GREENGARD AND Z. GIMBUTAS, *FMMLIB2D: A MATLAB toolbox for fast multipole method in two dimensions*, Version 1.2, 2012.
[14] L. GREENGARD AND M. MOURA, *On the numerical evaluation of electrostatic fields in composite materials*, Acta Numer., 3 (1994), pp. 379–410.
[15] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
[16] R. HAAS AND H. BRAUCHLI, *Fast solver for plane potential problems with mixed boundary conditions*, Comput. Methods Appl. Mech. Engrg., 89 (1991), pp. 543–556.
[17] ———, *Extracting singularities of Cauchy integrals—a key point of a fast solver for plane potential problems with mixed boundary conditions*, J. Comput. Appl. Math., 44 (1992), pp. 167–185.
[18] J. HELSING AND R. OJALA, *On the evaluation of layer potentials close to their sources*, J. Comput. Phys., 227 (2008), pp. 2899–2921.
[19] J. HELSING AND E. WADBRO, *Laplace's equation and the Dirichlet-Neumann map: a new mode for Mikhlin's method*, J. Comput. Phys., 202 (2005), pp. 391–410.
[20] P. HENRICI, *Applied and Computational Complex Analysis. Vol. 3*, Wiley, New York, 1986.
[21] P. KOEBE, *Abhandlungen zur Theorie der konformen Abbildung. IV. Abbildung mehrfach zusammenhängender schlichter Bereiche auf Schlitzbereiche*, Acta Math., 41 (1918), pp. 305–344.
[22] R. KRESS, *A Nyström method for boundary integral equations in domains with corners*, Numer. Math., 58 (1990), pp. 145–161.
[23] ———, *Linear Integral Equations*, 3rd. ed., Springer, New York, 2014.
[24] J. M. LEE, *Introduction to Topological Manifolds*, Springer, New York, 2000.
[25] Y. LIU, *Fast Multipole Boundary Element Method*, Cambridge University Press, Cambridge, 2009.
[26] A. I. MARKUSHEVICH, *Theory of Functions of a Complex Variable. Vol. II*, Prentice-Hall, Englewood Cliffs, 1965.
[27] S. G. MIKHLIN, *Integral Equations and their Applications to Certain Problems in Mechanics, Mathematical Physics and Technology*, Pergamon Press, New York, 1957.
[28] M. MOKRY, *Flow past airfoils as a Riemann-Hilbert problem*, in Theroretical Fluid Mechanics Conference New Orleans, 1996, AIAA paper 96-2161, AIAA, Boston, 1996.
[29] A. H. M. MURID AND M. NASSER, *Eigenproblem of the generalized Neumann kernel*, Bull. Malays. Math. Sci. Soc. (2), 26 (2003), pp. 13–33.
[30] N. I. MUSKHELISHVILI, *Singular Integral Equations*, Noordhoff, Leyden, 1977.
[31] M. NASSER, *A boundary integral equation for conformal mapping of bounded multiply connected regions*, Comput. Methods Funct. Theory, 9 (2009), pp. 127–143.
[32] ———, *Numerical conformal mapping via a boundary integral equation with the generalized Neumann kernel*, SIAM J. Sci. Comput., 31 (2009), pp. 1695–1715.
[33] ———, *The Riemann-Hilbert problem and the generalized Neumann kernel on unbounded multiply connected regions*, The University Researcher (IBB University Journal), 20 (2009), pp. 47–60.
[34] ———, *Boundary integral equations for potential flow past multiple aerofoils*, Comput. Methods Funct. Theory, 11 (2011), pp. 375–394.
[35] ———, *Numerical conformal mapping of multiply connected regions onto the second, third and fourth categories of Koebe's canonical slit domains*, J. Math. Anal. Appl., 382 (2011), pp. 47–56.
[36] ———, *Numerical conformal mapping of multiply connected regions onto the fifth category of Koebe's canonical slit regions*, J. Math. Anal. Appl., 398 (2013), pp. 729–743.

ETNA
Kent State University
http://etna.math.kent.edu

[37] ———, *Convergence of numerical solution of generalized Theodorsen's nonlinear integral equation*, Abstr. Appl. Anal., (2014), Art. ID 213296 (11 pages).
[38] ———, *Fast computation of the circular map*, Comput. Methods Funct. Theory, in press, doi: 10.1007/s40315-014-0098-3.
[39] M. NASSER AND F. AL-SHIHRI, *A fast boundary integral equation method for conformal mapping of multiply connected regions*, SIAM J. Sci. Comput., 35 (2013), pp. A1736–A1760.
[40] M. NASSER AND A. MURID, *Numerical experiments on eigenvalues of the generalized Neumann kernel*, in Advances in Group Theory, DNA Splicing and Complex Analysis, A. Murid and Y. Yaacob, eds., Penerbit UTM Press, 2012, pp. 135–158.
[41] M. NASSER, A. H. M. MURID, AND S. AL-HATEMI, *A boundary integral equation with the generalized Neumann kernel for a certain class of mixed boundary value problem*, J. Appl. Math., (2012), Art. ID 254123, (17 pages).
[42] M. NASSER, A. MURID, M. ISMAIL, AND E. ALEJAILY, *Boundary integral equation with the generalized Neumann kernel for Laplace's equation in multiply connected regions*, Appl. Math. Comput., 217 (2011), pp. 4710–4727.
[43] M. M. S. NASSER, A. H. M. MURID, AND A. W. K. SANGAWI, *Numerical conformal mapping via a boundary integral equation with the adjoint generalized Neumann kernel*, TWMS J. Pure Appl. Math., 5 (2014), pp. 96–117.
[44] M. M. S. NASSER, A. H. M. MURID, AND Z. ZAMZAMIR, *A boundary integral method for the Riemann-Hilbert problem in domains with corners*, Complex Var. Elliptic Equ., 53 (2008), pp. 989–1008.
[45] M. NASSER, T. SAKAJO, A. MURID, AND L. WEI, *A fast computational method for potential flows in multiply connected coastal domains*, Japan. J. Indust. Appl. Math., 32 (2015), pp. 205–236.
[46] S. T. O'DONNELL AND V. ROKHLIN, *A fast algorithm for the numerical evaluation of conformal mappings*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 475–487.
[47] A. RATHSFELD, *Iterative solution of linear systems arising from the Nyström method for the double-layer potential equation over curves with corners*, Math. Methods Appl. Sci., 16 (1993), pp. 443–455.
[48] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.
[49] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[50] I. VEKUA, *Generalized Analytic Functions*, Pergamon Press, London, 1992.
[51] R. WEGMANN, *Methods for numerical conformal mapping*, in Handbook of Complex Analysis: Geometric Function Theory. Vol. 2, R. Kühnau, ed., Elsevier, Amsterdam, 2005, pp. 351–477.
[52] R. WEGMANN, A. H. M. MURID, AND M. NASSER, *The Riemann-Hilbert problem and the generalized Neumann kernel*, J. Comput. Appl. Math., 182 (2005), pp. 388–415.
[53] R. WEGMANN AND M. NASSER, *The Riemann-Hilbert problem and the generalized Neumann kernel on multiply connected regions*, J. Comput. Appl. Math., 214 (2008), pp. 36–57.
[54] G. C. WEN, *Conformal Mappings and Boundary Value Problems*, Amer. Math. Soc., Providence, 1992.
[55] A. YUNUS, A. MURID, AND M. NASSER, *Numerical conformal mapping and its inverse of unbounded multiply connected regions onto logarithmic spiral slit regions and rectilinear slit regions*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 470 (2014), Art. ID. 20130514 (24 pages).
[56] ———, *Numerical evaluation of conformal mapping and its inverse for unbounded multiply connected regions*, Bull. Malays. Math. Sci. Soc. (2), 37 (2014), pp. 1–24.